

ВСТУП

База даних побудована за допомогою системи керування реляційними базами даних Microsoft SQL Server випуску 2008.

Microsoft SQL Server 2008 – це Microsoft SQL Server випуску 2008 року, вона є покращеною версією минулих випусків. В Microsoft SQL Server 2008 покращено систему захисту даних, оптимізовано працю системних функцій, локальних сховищ, веб-додатків та перероблено засіб допуску до даних.

Система SQL Server 2008 дозволяє звертатися до даних з усіх програм, розробленого із застосуванням технологій Microsoft .NET і Visual Studio, а також у межах сервісно-орієнтованої архітектури та бізнес-процесів – через Microsoft BizTalk Server.

Для розробки візуальної частини було використано інтегроване середовище для розробки програмного забезпечення Microsoft Visual Studio 2012.

Microsoft Visual Studio – лінійка середовищ розробки від компанії Microsoft, яка існує вже більше 9 років, слугує для розробки корпоративних додатків і веб-інтерфейсів з використанням .NET.

VisualStudio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторинга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і як відладчик машинного рівня. Решта вбудовувані інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу програми, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду додавання нових наборів інструментів.

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

РЕФЕРАТ

Об'єм роботи складає – 81 стор., кількість рисунків – 75, кількість використаної літератури – 6.

Об'єктом дослідження є база даних банку.

Метою і задачею проекту є структуризація і засвоєння, отриманих під час навчання, знань, навичок і умінь.

Методом дослідження є використання реляційної алгебри та теорії множин.

Курсовий проект є формою контролю отриманих та засвоєних знань з предмету «Організація баз даних».

Актуальність теми: визначається необхідністю розробки системи даних, що відноситься до теми проекту.

Предметом дослідження є концептуальне моделювання роботи системи відеопрокату.

Результатом проекту є розробка інформаційної системи відеопрокату.

Науковою новизною роботи є інтеграція в єдину базу даних прикладних систем баз даних, що описують різні аспекти діяльності систем банку.

Перелік ключових слів: БАЗА ДАНИХ, ТАБЛИЦЯ, ЗАПИТ, ПРОЦЕДУРА, ФУНКЦІЯ, ФІЛЬТР.

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ПЕРЕЛІК СКОРОЧЕНЬ, УМОНВИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

БД – база даних;
СБД – система баз даних;
СКБД – система керування баз даних;
ПКМ – права кнопка миші;
ЛКМ – ліва кнопка миші;

					5.151.1.42.08-ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Створення бази даних

Створимо базу даних за допомогою графічного інтерфейсу SQL Server Management Studio. Клацнемо правою кнопкою миші по контейнеру Database в Object Explorer і в контекстному меню виберемо New Database (Нова база). Відкриється діалогове вікно New Database, в лівій частині цього діалогового вікна бачимо три вкладки: General, Filegroups, Options.

На вкладці General задамо ім'я бази даних Library.

На вкладці Filegroups, визначимо, до якої файлової групі буде ставитися файл бази даних. Файлова група (Filegroup) - це спосіб організації файлів БД. За замовчуванням для будь-якої бази даних створюється файлова група PRIMARY, і все створювані файли будуть ставитися саме до неї. У створюваної БД все таблиці можна умовно поділимо на дві групи:

- призначені для користувача таблиці, які постійно змінюються користувачами;
- таблиці довідника, які змінюються дуже рідко.

Таким чином, при створенні БД створимо додаткову файлову групу USERS (вкладка Filegroups). Створимо новий файл даних USERS, і визначимо, що він буде ставитися до цієї файлової групі. Таблиці довідника залишимо в файлової групі PRIMARY.

1.1 Програмні продукти, що використовуються при проектуванні бази даних

Розроблено базу даних за допомогою програмного продукту Microsoft SQL Server 2008.

Microsoft SQL Server випуску 2008 - це надійна, продуктивна і інтелектуальна платформа даних, здатна відповідати потребам найбільш ресурсномістких бізнес-додатків. Вона дозволяє скоротити час і витрати на розробку і супровід додатків, а також надавати практично застосовну інформацію на кожне робоче місце підприємства.

Система SQL Server 2008 відштовхується від концепції платформи даних Майкрософт: вона спрощує управління будь-якими даними в будь-якому місці і в будь-який момент часу. Система дозволяє зберігати в базах даних інформацію, отриману з структурованих, напівструктурованих і неструктурованих джерел, таких як зображення і музика. У SQL Server 2008 є великий набір інтегрованих

					5.151.1.42.08-ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

служб, які розширюють можливості використання даних: можна складати запити, виконувати пошук, проводити синхронізацію, робити звіти, аналізувати дані. Всі дані зберігаються на основних серверах, що входять до складу центру обробки даних. До них здійснюється доступ з настільних комп'ютерів і мобільних пристроїв. Таким чином, користувач повністю контролює дані незалежно від того, де їх зберегли.

Система SQL Server 2008 дозволяє звертатися до даних з будь-якої програми, розробленої із застосуванням технологій Microsoft .NET і Visual Studio.

SQL Server забезпечує високий рівень безпеки, надійності і масштабованості для критично важливих додатків.

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудовуються інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1.2. Установка "MICROSOFT SQL SERVER 2008"

1. Откроется «Центр установки SQL Server» (SQL Server Installation Center), переходимо на вкладку «Установка» (Installation) і натискаємо на «Нова установка ізольованого примірника SQL Server або додавання компонентів до існуючої установки».

2. Запустітся «Програма установки SQL Server 2014» (SQL Server 2014 Setup). На першому кроці потрібно ввести ключ продукту (якщо він у вас є) або вибрати вільний випуск SQL Server. Це може бути:

- Evaluation Edition - повний набір компонентів, з обмеженням використання в 180 днів.

- Express Edition - безкоштовна версія з обмеженим функціоналом.

Залежно від введеного ключа, буде обрана для установки відповідає даному ключу редакція MS SQL Server 2014.

Ключ продукту можна буде ввести в будь-який час, після установки SQL Server. Про те як це зробити читайте в статті Зміна версії або введення ключа Microsoft SQL Server 2012.

Після вибору параметрів натискаємо «Далі» (Next).

3. Потім приймаємо умови ліцензійної угоди, погоджуємося (чи ні) на передачу даних про використання компонентів в корпорацію Microsoft. Тиснемо «Далі» (Next).

4. Убеждаемся що немає проблем, які можуть виникнути під час установки і, якщо все ОК, тиснемо «Далі» (Next). Якщо все ж є які-небудь проблеми, необхідно усунути їх і повторити процедуру перевірки натиснувши кнопку «Включити заново» (Re-run).

5. На наступному кроці пропонується перевірити наявність оновлень програми установки за допомогою центру служби Microsoft Update. Для цього встановлюємо прапор «Чи використовуєте цент служби Microsoft Update для перевірки наявності оновлень (рекомендовано)» і натискаємо «Далі».

					5.151.1.42.08-ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

6. Після скачування оновлень і установки файлів установки, переходимо безпосередньо до самої установки MS SQL Server 2014. Знову запускається перевірка необхідних вимог. Звернемо увагу на попередження програми про те, що Брандмауер Windows включений. Пізніше ми ще повернемося до цього питання. Якщо всі вимоги виконані, натискаємо «Далі» (Next), в іншому випадку добиваємося виконання всіх правил і запускаємо перевірку знову за допомогою кнопки «Включити заново» (Re-run).

7. Варіант установки вибираємо «Установка компонентів SQL Server» (SQL Server Feature Installaion) і натискаємо «Далі» (Next).

8. Тепер необхідно вибрати компоненти для установки. Для більшості випадків (наприклад для роботи і адміністрування кластера серверів 1С: Підприємство), цілком достатньо установки компонент:

- о Служби компонента Database Engine (Database Engine Services) - безпосередньо самої служби MS SQL Server.
- о Засоби управління - основні (Management Tools - Basic)

- о Засоби управління - повний набір (Management Tools - Complete) - і програми «Середовище SQL Server Management Studio» для адміністрування SQL Server

Всі компоненти завжди можна доставити / видалити, просто знову запустивши установку SQL Server. Вибравши необхідні компоненти, натискаємо «Далі» (Next).

9. У черговий раз запускається перевірка вимог для установки обраних на попередньому кроці компонент (набір вимог залежить від встановлюваних компонент). Наприклад, якщо заздалегідь не встановити платформу Microsoft .NET Framework 3.5, то побачите відповідну помилку. Запуск перевірки знову здійснюється за допомогою кнопки «Включити заново» (Re-run). Після задоволення всіх вимог натискаємо «Далі» (Next).

10. Тепер необхідно ввести ім'я іменованого екземпляра SQL Server або залишити ім'я екземпляра за замовчуванням. На одному ізольованому сервері можливо встановлювати до 50 іменованих примірників MS SQL Server 2014 і до 25 в відмов кластерів при використанні загального диска кластера (посилання). Примірник за замовчуванням (ідентифікатор MSSQLSERVER) може бути тільки один. Визначившись з ім'ям (Named instance) та ідентифікатором (Instance ID)

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

примірнику SQL Server або вибравши екземпляр за замовчуванням натискаємо «Далі» (Next).

11. Тепер виберемо режим запуску служб SQL Server. Тут можна задати обліковий запис Windows із під якою будуть запускатися різні служби SQL Server, вибравши її в стовпці «Ім'я облікового запису» (Account name) і ввівши пароль для цього облікового запису в стовпці «Пароль» (Password) навпроти відповідної служби.

У деяких випадках (наприклад, при необхідності звернення до мережесхресурсів) є сенс запускати Агента SQL (SQL Server Agent) з під доменної облікового запису.

12. Здесь необхідно визначитися з параметрами сортування баз даних за замовчуванням.

Для більшості програм, які будуть працювати з SQL сервером (наприклад для систем 1С: Підприємство) досить вибрати «Cyrillic_General_CI_AS». Указав параметри сортування натискаємо «Далі» (Next).

13. Тепер слід вибрати режим перевірки автентичності (Authentication Mode). Залежно від способу підключення до SQL Server можна вибрати «Режим перевірки автентичності Windows» (Windows authentication mode) або «Змішаний режим» (Mixed Mode) - в цьому випадку будуть використовуватися обидва режими.

У разі вибору «Змішаного режиму» (Mixed Mode) буде потрібно вести пароль на вбудовану обліковий запис «sa».

Тут же відразу можна додати адміністраторів SQL Server з числа наявних користувачів домену або локального комп'ютера за допомогою кнопок «Додати поточного користувача» (Add Current User) і «Додати ...» (Add ..).

Для того, щоб потім не забути введений при установці пароль, рекомендую використовуватися спеціальні менеджери паролів, наприклад безкоштовну програму KeePass.

Установивши усі параметри переходимо на вкладку «Каталоги даних» (Data Directories).

14. Перевіряємо всі введені на попередніх кроках параметри. Звернемо увагу, що значення всіх введених параметрів буде збережено в файлі конфігурації

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

ConfigurationFile.ini. При повторній установці SQL Server даний файл можна вказати як параметр для програми установки, в цьому випадку установка SQL Server пройде в автоматичному режимі, а настройки будуть взяті з конфігураційного файлу. Даний спосіб можна використовувати, якщо необхідно швидко встановити кілька екземплярів SQL Server з однаковими параметрами, або для «бекапа» параметрів прийнятих під час установки поточного екземпляра. Натискаємо «Встановити» (Install), для запуску процесу установки SQL Server.

15. Чекаємо закінчення установки, переконуємося, що всі компоненти встановлені успішно і завершуємо роботу майстра натиснувши на кнопку «Закрити» (Close).

16. Установка Microsoft SQL Server 2014 завершена.

2. Створення таблиць і заповнення

1. Всі таблиці БД знаходяться в папці "Tables" папки "Bank" у вікні оглядача об'єктів (рис.2.2.1).

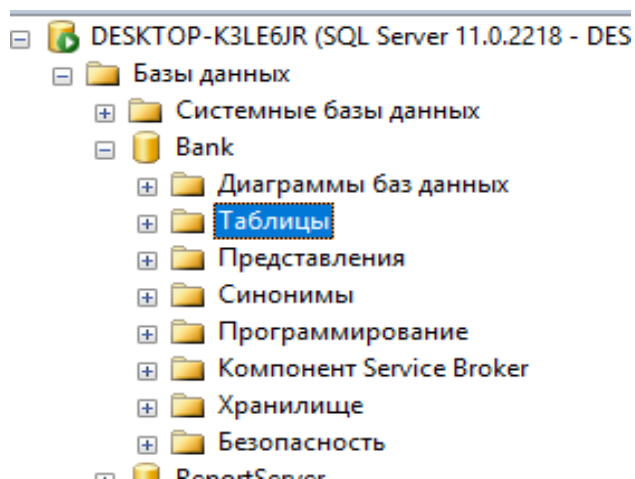


Рис.2.1- обозрівач БД з таблицями.

2. Створив таблиці "Workers1", "Pasitions", "Currency", "Contributors", "Contributions". Для цього щолкнул ПКМ по папці "Tables" і в меню вибрав пункт "New Table". З'явиться вікно створення нової таблиці (рис.2.2).

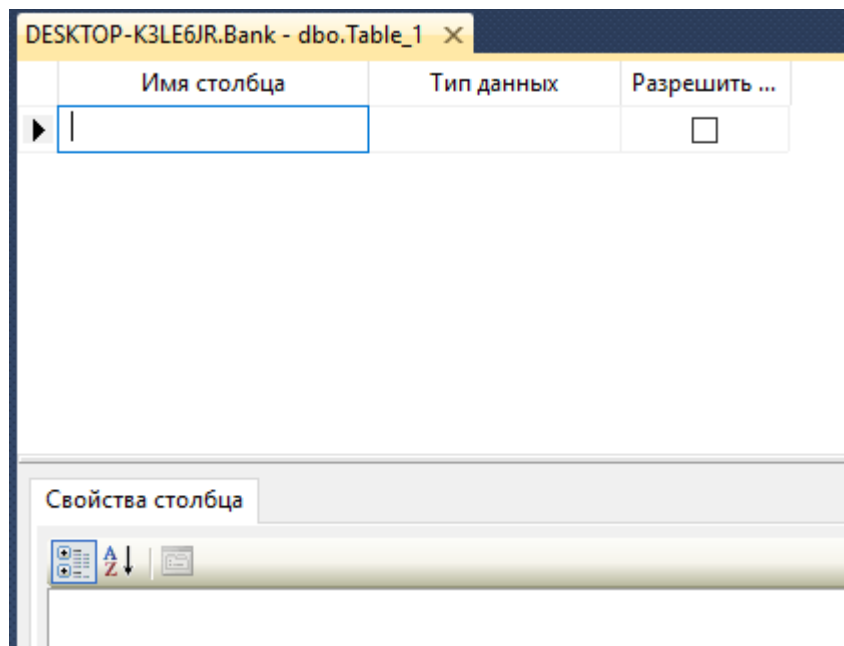


Рис.2.2- вікно створення таблиці.

3. У правій частині вікна розташована таблиця визначення полів нової таблиці. Дана таблиця має такі стовпці:

Column Name - ім'я поля. Ім'я поля має завжди починатися з літери і не повинно містити різних спеціальних символів н розділових знаків. Якщо ім'я поля містить прогалини, то воно автоматично полягає в квадратні дужки.

Data Type - тип даних поля.

Allow Nulls - допуск значення Null. Если ця опція поля включена, то в разі незаповнення поля в нього буде автоматично підставлений значення Null. Тобто поле необов'язкове для заповнення.

4. Опис спеціальності - текстове поле, призначене для зберігання рядків, мають необмежену довжину.

Зробив поле "Employee code" лічильником. Для цього виділіть поле, просто клацнувши по ньому мишкою в таблиці визначення полів. У таблиці властивостей поля

відобразяться властивості поля "Employee code". Розгорнув групу властивостей "Identity Specification" (Налаштування особливості). Властивість '(Is Identity)' (Особливе)

усіновіте в значення "Yes" (Так). Задав властивості "Identity Increment" (Збільшення

особливості, крок лічильника) і "Identity Send" (Початок особливості початкове значення лічильника) рівними 1.Ето настройки показують, що значення поля "Employee code" у першому записі в таблиці буде рівним 1, у другій 2, у третій 3 і т. д.

Потім зробимо поле "Код співробітника" ключовим полем. Виділив поле, а потім на панелі інструментів натиснув кнопку з зображення ключа.

У таблиці визначення полів, поруч з полем "Код спеціальності" з'явиться зображення ключа, що говорить про те, що поле ключове.

На цьому настройку таблиці "Workers1" можна вважати завершеною.Создав поля представлені на рисунку ниже (рис.2.3).

Имя столбца	Тип данных	Разрешить ...
[Employee code]	bigint	<input type="checkbox"/>
FIO	nvarchar(50)	<input type="checkbox"/>
Age	bigint	<input type="checkbox"/>
Gender	nchar(3)	<input type="checkbox"/>
Address	nvarchar(MAX)	<input type="checkbox"/>
Phone	varchar(30)	<input type="checkbox"/>
[Passport data]	varchar(MAX)	<input type="checkbox"/>
[Post code]	bigint	<input type="checkbox"/>
		<input type="checkbox"/>

Рис.2.3- вікно створенні полів таблиці 'Workers1'.

Закрило вікно створення нової таблиці, натиснувши кнопку закриття у верхньому правому

кутку вікна, над таблицею визначення полів. З'явиться вікно із запитом про збереження таблиці.

5. У новому вікні поставив ім'я нової таблиці як "Pasitions" і натиснув на кнопку "Ok". Таблиця "Pasitions" відобразиться в браузері об'єктів у програмі "Tables" БД "Bank".

Потім перейшов до створення таблиці "Currency". Як і у випадку з таблицею "Pasitions" клацнув ПКМ по папці "Tables" і в меню вибрав пункт "New Table". Створив поля вказані в завданні (рис.2.4).

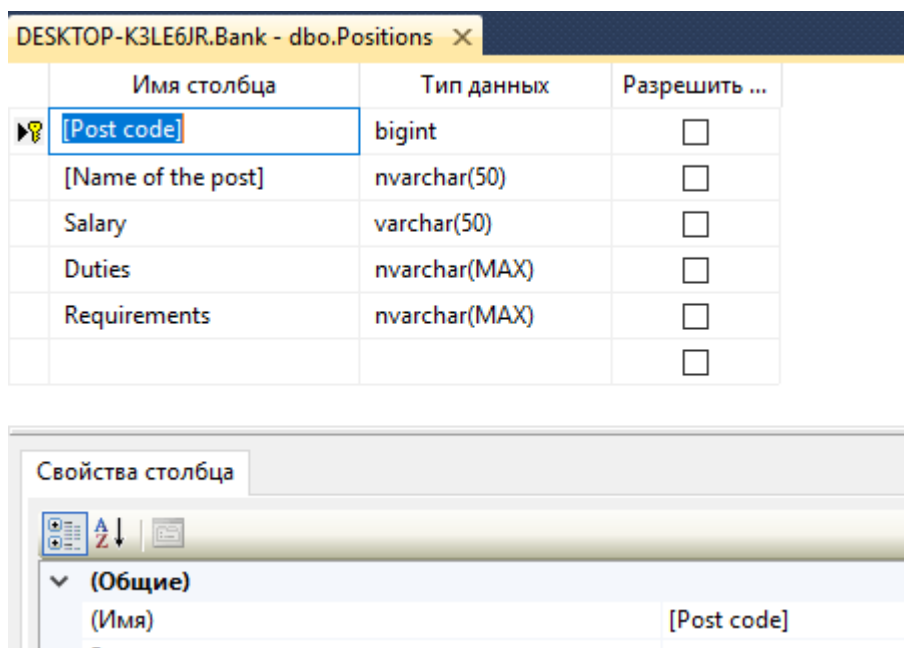


Рис.2.4- вікно створенні полей таблиці 'Positions'

6. Зробив поле "Currency code" числовим лічильником і ключовим полем, як це було зроблено в таблиці "Works1". Закрив вікно створення нової таблиці. У вікні "Chose Name" поставив ім'я "Currency" (рис.2.5).

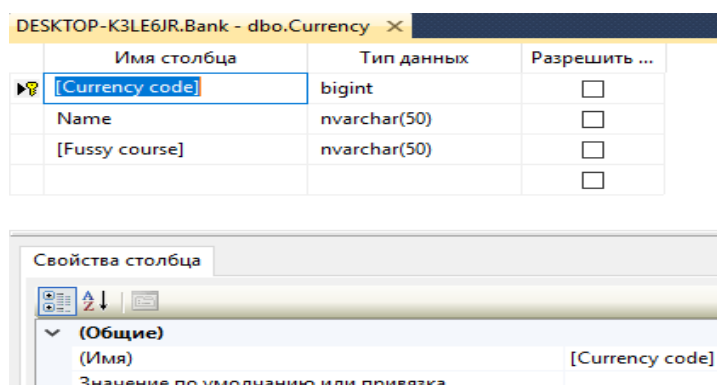


Рис.2.5- вікно створенні полей таблиці 'Currency'

Таблиця "Currency" з'явилася в папці "Tables" в браузері об'єктів. Після створення таблиці "Currency" створив таблицю "Contributors". Створив нову таблицю аналогічну таблиці "Works1" (рис.2.6).

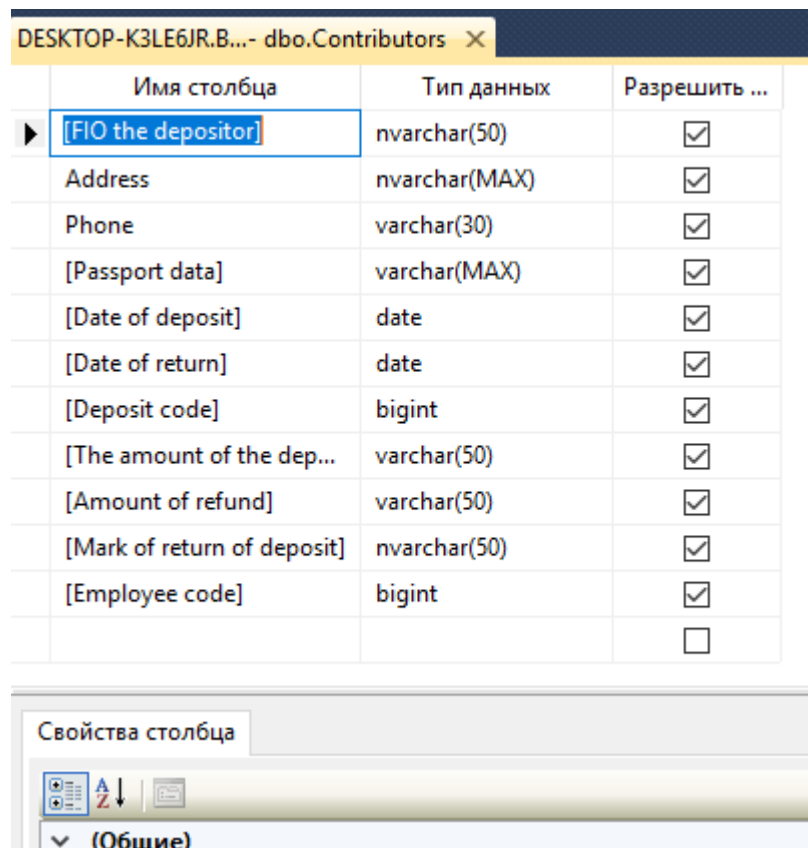


Рис.2.6- вікно створенні полів таблиці 'Contributors'

7. Поля "FIO", "Gender", "Address", "Age", "Phone", "Passport data",

"Mark of return of deposit" є текстовими полями різної довжини (для завдання довжини виділеного текстового поля необхідно в таблиці властивостей виділеного поля встановити властивість Length дорівнювала б максимальній кількості знаків тексту, що вводиться в поле);

Поля "Age" і "Date of return" і "Date of deposit" призначені для зберігання дат.

Тому вони має тп даних "date";

Поля "Passport data" і "Employee code" і "Currency code" є цілочисельними. Єдиною відмінністю є розмір полів. Поле "Passport data" призначено для зберігання

цілих чисел в діапазоні -2 (63) ... + 2 (64) (тп даних "bigint"). Поле "Курс" призначено для зберігання цілих чисел в діапазоні 0 ... 255 (тп даних "tinyint");

Поле "Employee code" - це поле зв'язку з таблицею "Contributors".

Однак, дане поле зв'язку є вторинним, тому його можна зробити просто цілочисельних, тобто, "bigint",

Після визначена полів таблиці "Contributors", закрив вікно створення нової таблиці. У вікні "Chose Name" задайте ім'я нової таблиці як

"Contributors".

Таблиця "Contributors" з'явиться в папці 'Table "в браузері об'єктів.

Нарешті, створимо таблицю "Contributions". Створив поля, вказані в завданні.

Таблиця "Contributions" не має первинних полів зв'язку. Отже, ця таблиця не має ключових полів. Поля "Employee code", є вторинним полем, призначеними для зв'язку з таблицею "Works1" тому вона є цілочисельним (тип даних 'bigint '). Поля "Date of return" і "Date of deposit" призначені для зберігання дат (тип даних "date "),(рис.2.7).

DESKTOP-K3LE6JR.B...dbo.Contributions X			
	Имя столбца	Тип данных	Разрешить ...
PK	[Deposit code]	bigint	<input type="checkbox"/>
	[Name of the deposit]	nvarchar(50)	<input type="checkbox"/>
	[Minimum deposit term]	nvarchar(50)	<input type="checkbox"/>
	[Minimum deposit amo...	nvarchar(50)	<input type="checkbox"/>
	[Currency code]	bigint	<input type="checkbox"/>
	[Interest rate]	varchar(50)	<input type="checkbox"/>
	[Additional conditions]	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Свойства столбца

Рис.2.7- вікно створенні полів таблиці 'Contributions'

8. Потім розглянув операцію заповнення таблиць початковими даними. Для початку заповнив таблицю "Workers1". Для заповнення цієї таблиці в браузері об'єктів клацнув правою кнопкою миші по таблиці "Workers1" і в меню вибрав пункт "Edit Top 200 Rows" (Редагувати перші 200 записів.). У робочій області "Microsoft SQL Server Management Studio" проявиться вікно заповнення таблиць. Заповніть таблицю "Workers1", як зазначено в завданні (рис.2.8).

	Employee code	FIO	Age	Gender	Address	Phone	Passport data	Post code
►	1	Броварчук Лю...	42	Жін	40388, Івано-Ф...	+380994986711	ЄД-0001021	1
	2	Крамарчук Іго...	24	Муж	22917, Запоріз...	+380672375770	ЄД-1010020	5
	3	Микитюк Вікто...	33	Муж	57569, Лугансь...	+380638589966	ЄД-1587631	2
	4	Пономарчук В...	27	Жін	75725, Хмельн...	+38(0448)808494	ЄД-1564664	1
	5	Алла Андрійов...	38	Жін	67558, Вінниць...	+38(057)9417387	МК-5476214	2
	6	Шевчук Анна Б...	32	Жін	71591, Дніпроп...	+380678372091	МК-4564515	1
	7	Віктор Анатолі...	51	Муж	60551, Чернігів...	+38(038)7102325	МС-5644557	2
	8	Антон Олексій...	28	Муж	82646, Лугансь...	+380965589349	ЄД-1565444	4
	9	Тарашук Інна ...	38	Жін	45220, Рівненс...	+380990305099	МК-5641324	3
	10	Юлія Олексійо...	45	Жін	07520, Львівськ...	+380671640544	МД-3567646	3

Рис.2.8- вікно заповнення таблиці 'Workers1'

9. Закрив вікно заповнення таблиці "Workers1" натиснувши кнопку закриття вікна в верхньому правому куті, над таблицею.

Після заповнення таблиці "Positions" заповнимо таблицю "Currency"(pic.2.2.9).

	Currency code	Name	Fussy course
►	1	UAH - гривня	26.90 UAH = 1 ...
	2	USD -долар	1 USD = 26.90 U...
	3	EUR - євро	1 EUR = 1.13 USD

Рис.2.9- вікно заповнення таблиці "Currency"

Відкрийте її для заповнення як описано вище.

Закрив вікно заповнення таблиці "Currency" і перейдіть до заповнення таблиці "Contributions", Відкрийте таблицю "Contributions" для заповнення і заповнив її як зазначено в завданні (рис.2.10).

	Deposit code	Name of the d...	Minimum dep...	Minimum dep...	Currency code	Interest rate	Additional con...
►	121	Срочний депо...	90 дней	от 1 300 UAH	1	15,00%	Виплата проце...
	122	Срочний депо...	90 дней	от 1 300 UAH	1	14,75%	Виплата проце...
	142	Срочний депо...	90 дней	от 100 USD	2	2,75%	Виплата проце...
	154	Срочний депо...	90 дней	от 100 EUR	3	0,50%	Виплата проце...
	221	Депозитний се...	3 місяця	от 1 300 UAH	1	15,50%	Виплата проце...
	242	Депозитний се...	3 місяця	от 100 USD	2	3,25%	Виплата проце...

Рис.2.10- вікно заповнення таблиці 'Contributions'

Для заповнення дат як роздільник міллю використовувати знак ".". Дати морю заповнювати в форматі "день.місяць.рік".

Поле "Employee code" є вторинним полем зв'язку (для згідно з таблицею "Workes1"). Отже, значення цього поля необхідно заповнювати значеннями поля "Employee code" таблиці "Workes1". В цьому випадку це значення від 1 до 5. Нарешті заповнив таблицю "Contributors"(рис.2.11).

FIO the deposi...	Address	Phone	Passport data	Date of deposit	Date of return	Deposit code	The amount o...	Amount of ref...	Mark of return...	Employee code
Кравченко Світ...	72348, Донецьк...	0636609769	MN178647	2019-02-25	2019-05-25	121	5 000 UAH	5 182,5 UAH	Повернено	1
Кравченко Світ...	83049, Херсонс...	+380934492577	MK464564	2019-02-27	2019-05-27	122	10 000 UAH	10 125 UAH	Повернено	1
Крамаренко Ф...	48081, Херсонс...	+380506999891	BC456512	2018-05-15	2019-05-15	121	25 000 UAH	28 812 UAH	Повернено	2
Анастасія Евге...	49676, Миколаї...	0638018942	EB245368	2019-01-13	2020-01-13	142	1 000 USD	1 040 USD	Не повернено	1
Іванченко Сер...	44766, Миколаї...	0936382162	MN985621	2017-10-19	2019-10-19	122	50 000 UAH	50 656,25 UAH	Повернено	2
Дмитренко Ти...	98202, Хмельн...	+380632882863	MN564312	2019-01-05	2020-01-05	154	1 000 EUR	1 010 EUR	Не повернено	2
Микитюк Мир...	83154, Львівськ...	+380999512901	BE262375	2018-08-03	2019-08-03	242	10 000 USD	10 500 USD	Повернено	1
Крамаренко Га...	99982, Черкась...	0997160618	MN254687	2018-08-08	2020-08-08	121	60 000 UAH	79 200 UAH	Не повернено	2
Юлія Васильов...	75833, Одеська...	0674171211	MN387962	2017-03-28	2020-03-28	221	150 000 UAH	226 500 UAH	Не повернено	4
Середа Сергій ...	92889, Чернігів...	0505643449	MK576248	2018-01-25	2019-01-25	221	50 000 UAH	58 500 UAH	Повернено	1

Рис.2.11- вікно заповнення таблиці 'Contributors'

На цьому я закончив заповнення таблиць.

3. Створення запитів і фільтрів

1. Перешол до створення статичних запитів. У браузері об'єктів "Microsoft SQL Server 2008" все запити БД знаходяться в папці "Views". Створимо запит "Запит Department of Caldas", що зв'язує таблиці "Workers1" і "Pasitions" по полю зв'язку "Post code". Для створення нового запиту необхідно в браузері об'єктів в БД "Bank" клацнути ПКМ по папці "Views", потім в меню вибрати пункт "New View". з'явиться вікно "Add Table" (Додати таблицю), призначене для вибору таблиць і запитів, які беруть участь в новому запиті (рис.3.1).

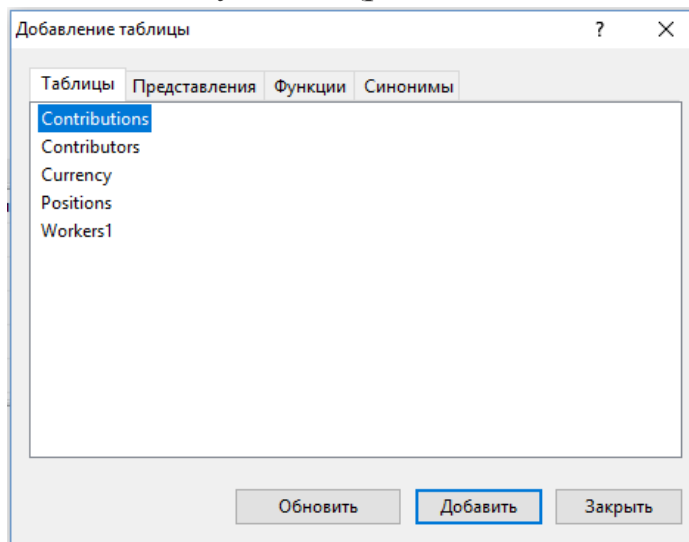


Рис.3.1-вікно створення фільтру.

2. Додамо в новий запит таблиці "Workers1" і "Pasitions". Для цього у вікні "Add Table" виділив таблицю "Workers1" і натисніть кнопку "Add" (Додати). Аналогічно додайте таблицю "Pasitions". Після додавання таблиць беруть участь в

запиті закрити вікно "Add Table" натиснувши кнопку 'Close' (Закрити). З'явиться вікно конструктора запитів. Вікно конструктора запитів складається з наступних панелей (рис.3.2):

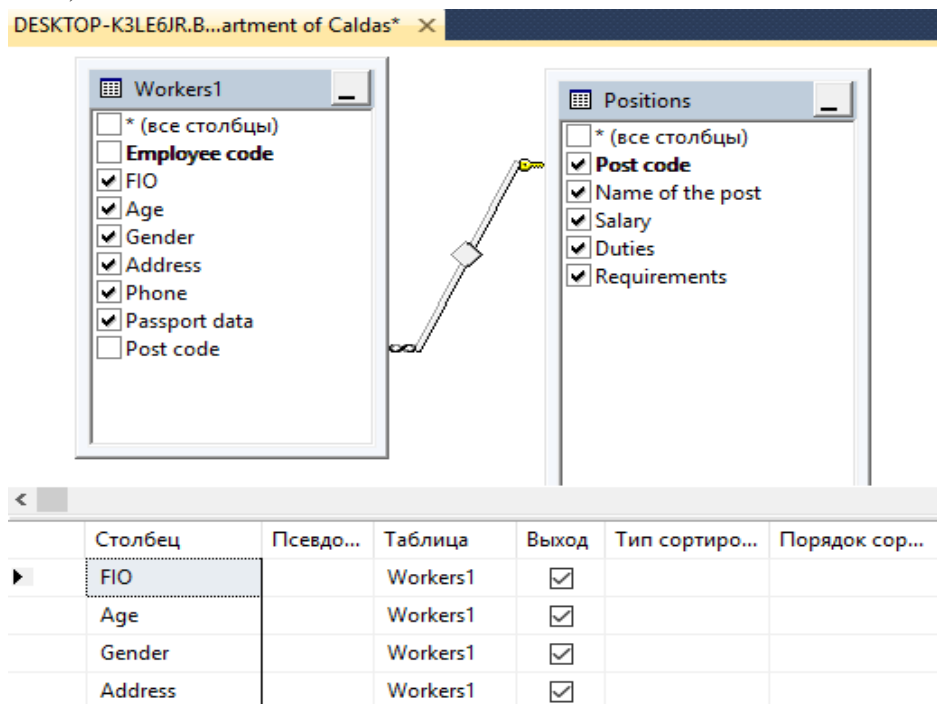


Рис.3.2-запит 'Department of Caldas'.

1) Схема даних - відображає статі таблиць і запитів, які беруть участь запиті, дозволяє вибирати відображаються статі, дозволяє встановлювати зв'язки між учасниками запиту по спеціальних лініях зв'язку. Ця панель включається і виключається наступною кнопкою на панелі інструментів.

2) Таблиця видимі поля - показує відображаються поля (стовпець "Column"), дозволяє задати їм псевдоніми (стовпець Alias "), дозволяє встановлювати тип сортування записів по одному або декількох полях (стовпець " Sort Type '), дозволяє ставити порядок сортування (стовпець "Sort Order"), дозволяє задавати умови відбору записів в фільтрах (стовпці "Filter" і "Or ..."). Також ця таблиця дозволяє змінювати порядок відображення полів у запиті. Ця панель вмикається та вимикається наступною кнопкою на панелі інструментів.

2) Код SQL - код створюваного запиту на мові T-SQL. ця панель включається н вимикається кнопкою на панелі інструментів.

4) Результат - показує результат запиту після його виконання. ця панель вмикається і вимикається кнопкою на панелі інструментів

Якщо необхідно знову відобразити вікно "Add Table 'для додавання нових таблиць або запитів, то для цього з панелі інструментів "Microsoft SQL

Sefver 2008 "потрібно натиснути кнопку натбіца.

Якщо необхідно видалити таблицю або запит із схеми даних, то для цього потрібно клацнути ПКМ н в меню вибрали пункт "Remove" (Вилучити).

Потім перейшов до зв'язування таблиць "Pasitions" і "Pasitions" по полам зв'язку "Post code". Щоб створити зв'язок необхідно в схемі даних перетягнути мишею поле "Post code" таблиці "Pasitions" на таке ж поле таблиці "Pasitions". Зв'язок відобразилася у вигляді ламаної лінії що з'єднує ці два поля зв'язку.

Якщо необхідно видалити зв'язок, то для цього необхідно клацнути по ній ПКМ н в меню вибрати пункт "Remove".

Після зв'язування таблиць (в такі при будь-яких поставленні в запиті) в області коду T-SQL буде відображатися T-SQL код редагованого запиту.

Тепер визначимо поля, які відображаються при виконанні запиту відображаютьсяполя позначаються плоскою (зліва від імені поля) на схемі даних, а такожвідображаються в таблиці видимі поля. Щоб зробити поле відображуваним при виконанні запиту необхідно зафіксувати покажчик миші по порожньому вапрос (зліва від імені статі) на схемі даних, в квадраті з'явиться галочка.

Якщо необхідно зробити поле невидимим при виконанні запиту, то потрібно прибрати галочку, розташовану зліва від імені статі на схемі даних. Для цього просто клацніть мишею по галочці.

Якщо необхідно відобразити всі поля таблиці, то необхідно встановити галочку зліва від пункту "*" (All Columns) (Всі поля), що належить відповідній таблиці на схемі даних.

Визначив відображаються поля нашого запиту (Отобраваотса все статі крім полів з кодами, тобто полів зв'язку).

На цьому настройку нового запиту можна вважати закінченою. перед збереженням запиту перевірив його працездатність, виконавши його. Для запуску апиту з панелі інструментів натиснув кнопку F5.

Або клацніть ПКМ в будь-якому місці вікна конструктора запитів і в меню оберіть пункт "Execute SQL". Результат виконуються запиту з'явиться у вигляді таблиці в області результату.

Якщо після виконання запиту результат не з'явився, а з'явилося повідомлення про помилку, то в цьому випадку перевірте, чи правильно створивши зв'язок.

Ламана лінія зв'язку долина з'єднувати поля "Post code" в обох таблицях.

4. Якщо лінія зв'язку з'єднує інші поля, то її необхідно видалити і створити заново, як це описано вище.

Якщо запит виконується правильно, то необхідно зберегти. Для збереження запиту закрив вікно конструктора запитів, щокнував мишкою по кнопці закриття розташованій в верхньому правому куті вікна конструктора (над схемою даних) з'явиться вікно з питанням про збереження запитів.

5. В появившемся вікні натиснув кнопку "Yes" (Так). з'явилася вікно "Choose Name" (Виберіть ім'я).

6. У новому вікні поставив ім'я нового запиту "Department of Caldas" і натиснув на кнопку "Ok". Запит з'явиться в папці "Views" БД "Bank" в браузері об'єктів (рис.3.3).

	FIO	Age	Gender	Address	Phone	Passport data	Name of the p...	Salary	Duties	Requirements	Post code
▶	Дмида Євгенович	42	Жін	40388, Івано-Ф...	+380994986711	ЄД-0001021	Касир-операці...	8500 грн.	Касир-операці...	великий тягар ...	1
	Крамарчук Іго...	24	Муж	22917, Запоріж...	+380672375770	ЄД-1010020	Начальник уп...	10000 грн.	Обязанности: ...	Опыт работы: ...	5
	Микитюк Вікто...	33	Муж	57569, Лугансь...	+380638589966	ЄД-1587631	Менеджер	9000 грн.	Менеджер з кр...	Освіта: вища / ...	2
	Пономарчук В...	27	Жін	75725, Хмельн...	+38(0448)808494	ЄД-1564664	Касир-операці...	8500 грн.	Касир-операці...	великий тягар ...	1
	Алла Андрійов...	38	Жін	67558, Вінниць...	+38(057)9417387	МК-5476214	Менеджер	9000 грн.	Менеджер з кр...	Освіта: вища / ...	2
	Шевчук Анна Б...	32	Жін	71591, Дніпроп...	+380678372091	МК-4564515	Касир-операці...	8500 грн.	Касир-операці...	великий тягар ...	1
	Віктор Анатолі...	51	Муж	60551, Чернігів...	+38(038)7102325	МС-5644557	Менеджер	9000 грн.	Менеджер з кр...	Освіта: вища / ...	2
	Антон Олексій...	28	Муж	82646, Лугансь...	+380965589349	ЄД-1565444	Директор відді...	15000 грн.	Організація пр...	повна вища ос...	4
	Тарашук Інна ...	38	Жін	45220, Рівненс...	+380990305099	МК-5641324	Фінансист	10000 грн.	Фінансист-роз...	аналітичний с...	3
	Юлія Олексійо...	45	Жін	07520, Львівськ...	+380671640544	МД-3567646	Фінансист	10000 грн.	Фінансист-роз...	аналітичний с...	3

Рис.3.3- запит "Department of Caldas".

7. Перевіримо працездатність створеного запиту поза конструктора запитів.

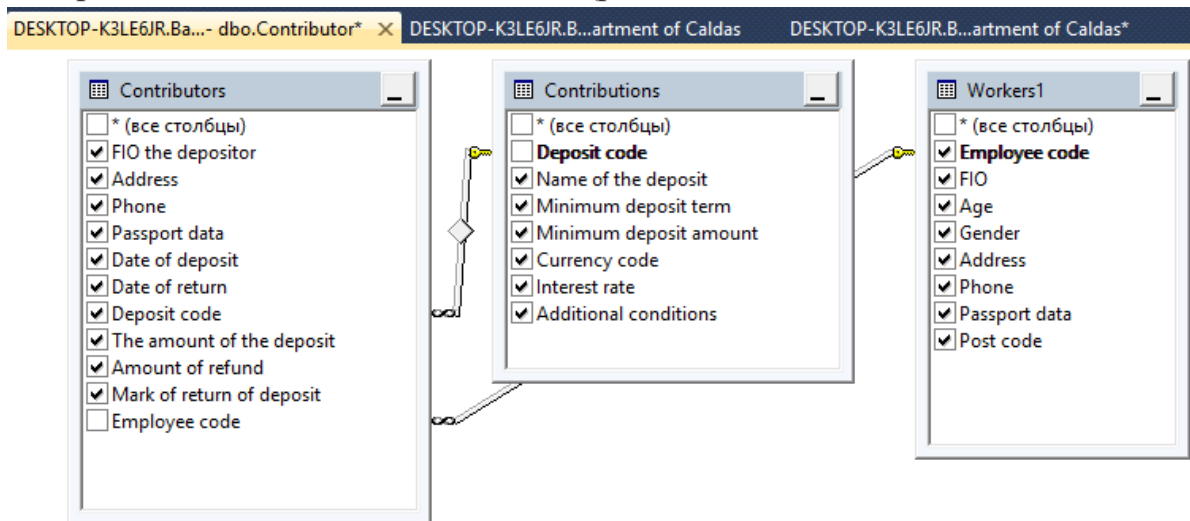
Запуску новостворений запит "Department of Caldas" без використання конструктора запитів. Для виконання вже записаний запит необхідно клацнути ПКМ за запитом і в меню вибрати пункт "Select top 1000 rows" (Відобразити перші 1000 записів). Виконав цю операцію для запиту "Department of Caldas".

8. Перешол до створення запиту "Cintribution". У браузері об'єктів в БД "Bank" щелкнути ПКМ по папці "Views", потім в меню вибрати пункт "New View". З'явилася вікно "Add Table".

У запиті "Cintribution" я пов'язав таблиці "Currency" і "Contributions" по полях зв'язку "Currency code". Отже, у вікні "Add Table" у новий запит додав таблиці "Currency" і "Contributions".

Після доавлення таблиць закрив вікно, з'явилася вікно конструктора запитів.

У вікні конструктора запитів встановив зв'язку між таблицями і визначив відображаються поля, як показано на (рис.3.4).



	Столбец	Псевдо...	Таблица	Выход	Тип сортиро...	Порядок сор...	Фильтр	Или...	Или...	Или...
►	[FIO the depo...		Contribut...	<input checked="" type="checkbox"/>						
	Address		Contribut...	<input checked="" type="checkbox"/>						
	Phone		Contribut...	<input checked="" type="checkbox"/>						
	[Passport data]		Contribut...	<input checked="" type="checkbox"/>						
	[Date of depo...		Contribut...	<input checked="" type="checkbox"/>						
	[Date of return]		Contribut...	<input checked="" type="checkbox"/>						
	[The amount ...		Contribut...	<input checked="" type="checkbox"/>						
	[Amount of r...		Contribut...	<input checked="" type="checkbox"/>						

Рис.3.4- запит "Cintribution".

Потім поміняв порядок видимі поля в запиті, для цього в таблиці видимі поля необхідно перетягнути поля мишею вгору або вниз за заголовок рядка таблиці (стовпець перед стовпцем "Column"). Розположена відображаються поля в таблиці як зазначено в завданні.

9. Задав псевдоніми для кожного з полів, просто записавши псевданіми в стовпці "Alias" таблиці видимі поля.

Перевірив працездатність нового запиту, виконуючи его. Обрав увагу на те, що реальні назви полів були замінені їх псевданімами. Закрив вікно конструктора. У появішемся вікні "Choose Name" поставив ім'я нового запиту "Cintribution".

Перевірив робото спроможність нового запиту поза конструктора. Для цього запустив запит.

10. Перешол до створення запиту "Contributor". У браузері об'єктів в БД "Bank" щелкнути ПКМ по папці "Views", потім в меню вибираємо пункт "New View". З'явилося вікно "Add Table".

У запиті "Contributor" я пов'язав таблиці "Workers1", "Contributors" і "Contributions." "Contributors" і "Contributions" зв'язав по полях зв'язку "Deposit code". А "Workers1" і "Contributors" зв'язав по полях зв'язку "Employee code".

Після дабовлення таблиць закрити вікно, з'явилося вікно конструктора запитів.

У вікні конструктора запитів встановив зв'язку між таблицями і визначив відображаються поля.

Потім поміняв порядок видимі поля в запиті, для цього в таблиці видимі поля необхідно перетягнути поля мишею вгору або вниз за заголовки рядка таблиці (стовпець перед стовпцем "Column"). Розположена відображаються поля в таблиці як зазначено в завданні.

Задав псевдоніми для кожного з полів, просто записавши псевдоніми в стовпці "Alias" таблиці видимі поля.

Перевірив працездатність нового запиту, виконуючи його. Обрав увагу на те, що реальні назви полів були замінені їх псевдонімами. Закрити вікно конструктора. У появившемся вікні "Choose Name" поставив ім'я нового запиту "Contributor".

Перевірив робото спроможність нового запиту поза конструктора. Для цього запустив запит.

11. Создание фільтрів. На основі запиту "Department of Caldas", створив фільтр, що відображає "Name of the post". У появившемся вікні конструктора запитів визначив як видимі поля всі поля запитів "Department of Caldas".

Для відображення всіх полів запиту, в даному випадку, ми не можемо використовувати пункт "*" (All Columns) (Всі статті). Так як в цьому випадку ми не можемо встановлювати критерій відбору записів в фільтрі, а також неможливо встановити сортування записів.

Потім встановив критерій відбору записів в фільтрі. Нехай фільтр відображає кожного працівника з однаковими роботами разом, а решта по окремих. Для визначення умови відбору записів в таблиці видимі поля в рядку, що відповідає полю, на яке накладається умова, в стовпці "Filter", необхідно задати умову. У моєму випадку умова накладається на поле "Name of the post". Отже, в рядку "Name of the post", в стовпці "Filter" потрібно задати наступну умову відбору - "N"Касір-операціоніст"(рис.3.5).

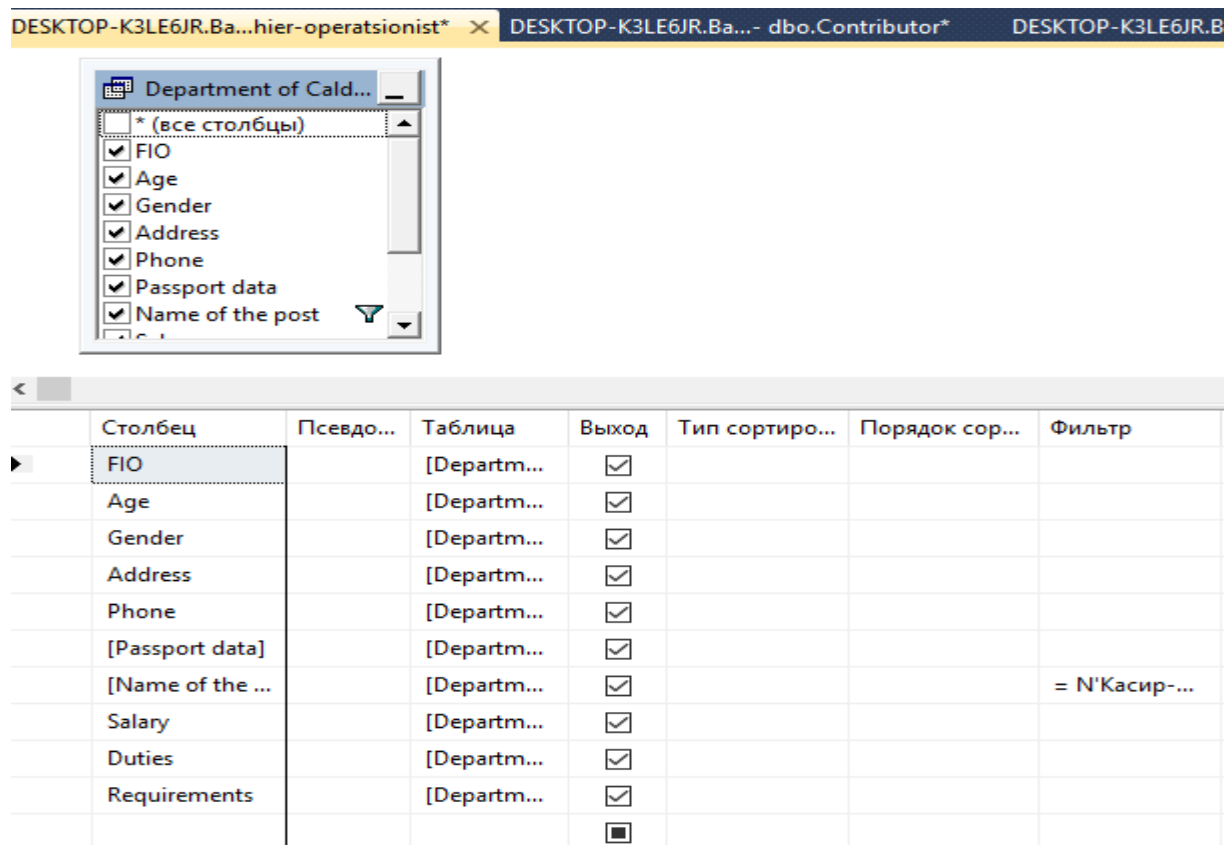


Рис.3.5-фільтр для запиту "Department of Caldas".

12. Аналогічно створив фільтри для відображення інших робочих місць.

Дані фільтри створюється аналогічно фільтру "Фільтр = N'Касир-операціоніст" (смотеть вище).

Єдиною відмінністю є умова відбору накладається на поле "Name of the post", воно повинно бути не "= N'Діректор відділення банку ", а "= N'Фінансіст ", "= N'Менеджер" і тд. При збереженні фільтрів задав їх імена їх умовам відбору, тобто "Filter for employees Cashier-operatsionist", "Filter for employees Director", "Filter for employees Financier", "Filter for employees Manager", "Filter for employees Security". Перевірів створені фільтри на працездатність .

13. Тепер на основі запиту "Contributor" створимо фільтри, отображающие серийный код депозиту. У таблиці видимі поля в солбце "Deposit code", встановив умова відбору рівне "= '121' ". Дані фільтри створюється аналогічно фільтру "= '121' " (смотеть вище).

Єдиною відмінністю є Умова відбору накладається на поле "Deposit code", воно винне бути "= '122'", "= '142'", "= '154'", "= '221'", "= '242'" і тд. При збереженні

фільтрів задавши їх імена їх умів відбору, тобто "Deposit account with separate deposits 121", "Deposit account with separate deposits 122", "Deposit account with separate deposits 142", "Deposit account with separate deposits 154", "Deposit account with separate deposits 221 ", " Deposit account with separate deposits 242 ". Перевірив створені фільтри на працездатність.

14. Створивши останні фільтри для відображення позначки про повернення вкладу. Даний фільтр створює також як і Попередні. Єдина відмінність являється умова відбору, накладуемое на поле ЕЕ, воно виглядає так «= N'Повернено'», а « = N'Не повернено'» . При збереженні фільтри поставив їм імена «Deposit Return Filter Returned », і «Deposit return filter is not returned».

На цьому я завершив створення фільтрів.

4.Процедури, що зберігаються

1. Перешол до створення збережених процедур. Для роботи з збереженими процедурами в браузері об'єктів. Створив процедуру, яка обчислює середнє трьох чисел. Для створення нової процедури, що щолкнул ПКМ по папці 'Stored Procedures' і в появішемся меню вибрав пункт 'New Stored Prochure'. З'явився вікно коду нової процедури.

2. 1) Область настройки параметрів синтаксису процедури. Дозволяє настрашити деякі синтаксичні правила, використовувані при наборі коду процедури. У нашому випадку це;

SET ANSI_NULLS ON - включає використання значень NULL (Пусто) в кодуванні ANSI,

SET QUOTED_IDENTIFIER ON - включає можливість використання подвійних лапок для визначення ідентифікаторів;

2) Область визначення імені процедури (Procedure_Name) н параметрів переданих в процедуру (@ Param1, @ Param2). Визначення параметрів має наступний синтаксис:

@ <Імя параметра> <Тип даних> = <Значення за замовчуванням>

Параметри розділяється між собою комами;

3) Початок тіла процедури, позначається службовим словом "BEGIN";

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

- 4) Тіло процедури, містить команди мови програмування запитів T-SQL;
- 5) Кінець тіла процедури, позначається службовим словом 'END ',

У коді зеленим кольором виділяється коментарі. Він не обробляється сервером і виконують функцію пояснень до коду, Рядки коментарів починаються з підрядка "--". Далі в коді, ми не будемо відобразити коментарі, вони будуть згорнуті. Зліва від розділу з коментарями буде спити знак "+", клацнувши по якому можна розгорнути коментар.

обробляється сервером і виконують функцію пояснень до коду, Рядки коментарів починаються з підрядка "--". Далі в коді, ми не будемо відобразити коментарі, вони будуть згорнуті. Зліва від розділу з коментарями буде спити знак "+", клацнувши по якому можна розгорнути коментар.

3. 1) CREATE PROCEDURE [Середнє трьох величин] - визначає ім'я створюване процедури як 'Середнє трьох величин';

2) @Value Real = 0, @ Value2 Real = 0, @ Value3 Real = 0 - визначають три параметра процедури Value1, Value2 і Value3. Даним параметрам можна привласнити дробові числа (Тип даних Real), значення за замовчуванням рівні 0;

3) SELECT 'Середнє значення' = (@ Value1 + @ Value2 + @ Value3) / 3 - обчислює

середнє і виводить результат з підписом "Середнє значення '.

Решта фрагменти кадра розглянуті вище.

Для створення процедури, виконаємо вищеописаний код, натиснувши кнопку F5, (Виконати) на панелі інструментів. У нижній частині вікна з кодом з'явиться повідомлення 'Command (s) completed successful, ". Закрийте вікно з кодом, клацнувши мишею по кнопці закриття розташованої у верхньому правому куті вікна з кодом процедури. Перевіримо працездатність створеної збереженої процедури. Для запуску процедури, що необхідно створити новий порожній запит, натиснувши на кнопку

(Новий запит) на панелі інструментів. У вікні з порожнім

запитом прибрав команду EXEC [Середнє трьох величин] 1, 7, 9 і натисніть кнопку Execute.

4. У нижній частині вікна з кодом з'явився результат виконання нової процедури, що: Середнє значення 5,666667.

Тепер створимо збережену процедуру для відбору студентів з таблиці

студенти по їх "ПІБ". Для цього створіть нову збережену процедуру, як це описано вище, і набрав код нової процедури.

5. Розглянув код процедури "Відображення працівники по ПІБ більш докладно:

					5.151.1.42.08-ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

1) CREATE PROCEDURE [Відображення працівники по ПІБ] - визначає ім'я створеної процедури як "Відображення працівники по ПІБ";

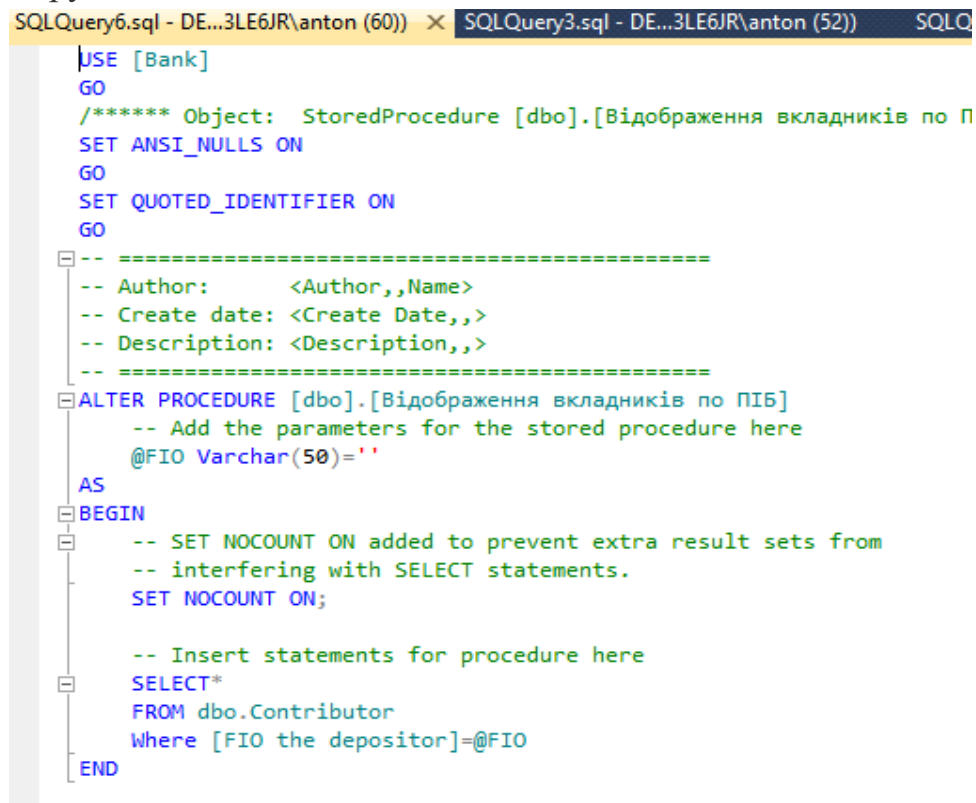
2) @ FIO Varchar (50) = "- визначають єдиний параметр процедури FIO.

Параметру можна привласнити текстові рядки змінної довжини, довжиною до 50 символів (Тип даних Varchar (50)), значення за замовчуванням рівні порожній рядку;

3) SELECT * FROM dbo.Студенти WHERE ПІБ = @ FIO - відобразить всі статі (*) з таблиці студенти (dbo.Студенти), де значення статі ПІБ дорівнює значенню параметра FIO (ПІБ = @ FIO).

Виконаємо вищеописаний код і закриємо вікно з кодом, як описано вище.

Перевірів працездатність створеної процедури, що, Створив новий порожній запит. У вікні з порожнім запасом наберіть команду EXEC [Відображення працівників по ПІБ] 'Броварчук Людмила Євгенович' н натиснув кнопку Execute на панелі інструментів.



```
SQLQuery6.sql - DE...3LE6JR\anton (60)) X SQLQuery3.sql - DE...3LE6JR\anton (52)) SQLQ
USE [Bank]
GO
/***** Object: StoredProcedure [dbo].[Відображення вкладників по П
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[Відображення вкладників по ПІБ]
-- Add the parameters for the stored procedure here
@FIO Varchar(50)=''
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
SELECT*
FROM dbo.Contributor
Where [FIO the depositor]=@FIO
END
```

Рис.4.1- Створення процедури.

DESKTOP-K3LE6JR.Bank - dbo.Workers1 SQLQuery7.sql - DE...3LE6JR\anton (62))* SQLQuery6.sql - DE...3LE6JR\anton (60)) SQLQuery3.sql - DE...3LE6JR\anton (52)

еxес [Відображення працівників по ПІБ] 'Броварчук Людмила Євгенович'

100 %

Результаты Сообщения

	Employee code	FIO	Age	Gender	Address	Phone	Passport data	Post code
1	1	Броварчук Людмила Євгенович	42	Жін	40388, Івано-Франківська область, місто Івано-Фр...	+380994986711	ЄД-0001021	1

Рис.4.2-Вивід процедури.

6. У нижній частині вікна з кодом з'явився результат виконання процедури "Оображення працівників по ПІБ".

Потім перейшов до більш складної задачі - відобразити вкладників по депозит коду. Створіть нову збережену процедуру на набрав код:

SQLQuery3.sql - DE...3LE6JR\anton (52)) SQLQuery1.sql - DE...3LE6JR\anton (56))

```

USE [Bank]
GO
/***** Object: StoredProcedure [dbo].[Відображення працівників по ПІБ]    Scr
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:        <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[Відображення працівників по ПІБ]
-- Add the parameters for the stored procedure here
@FIO Nvarchar(50)=''
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
SELECT*
FROM dbo.Workers1
Where FIO=@FIO
END

```

Рис.4.3- Створення процедури.

DESKTOP-K3LE6JR.B...- dbo.Contributors DESKTOP-K3LE6JR.Ba...- dbo.Contributor DESKTOP-K3LE6JR.Bank - dbo.Workers1 SQLQuery7.sql - DE...3LE6JR\anton (62))* SQLQu

еxес [Відображення вкладників по ПІБ] 'Кравченко Світлана Валентинович'

100 %

Результаты Сообщения

	FIO the depositor	Address	Phone	Passport data	Date of deposit	Date of return	The amount of the deposit	A
1	Кравченко Світлана Валентинович	83049, Херсонська область, місто Херсон, пл. Вол...	+380934492577	МК464564	2019-02-27	2019-05-27	10 000 UAH	1

Рис.4.4- Вивід процедури.

7. Але це я закінчив роботу над збереженими процедурами і переходом до розглядення призначених для користувача функцій.

5. Пользовательские функции

Рассмотрел створення і застосування призначених для користувача функцій. В БД 'Microsoft SQL Server 2008' всі призначені для користувача функції знаходяться в папці 'Functions' розташованій в папці 'Programmability' в браузері об'єктів.

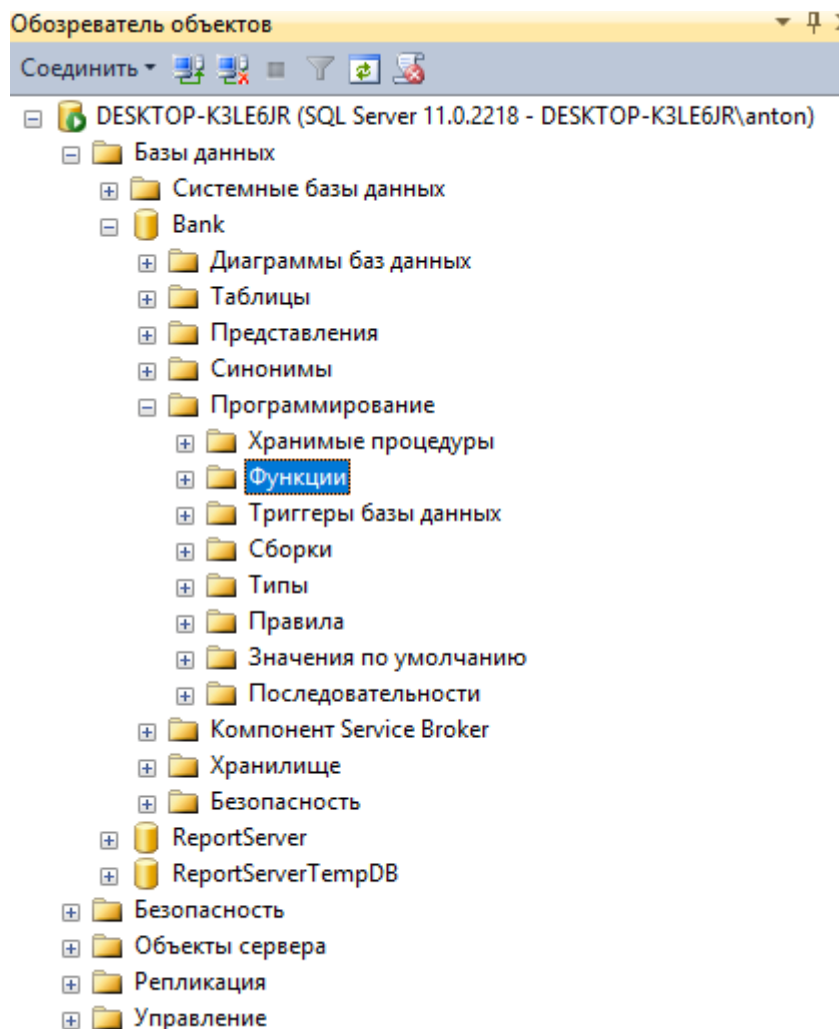


Рис.5.1-розположення функцій.

Почав з створення скалярних призначених для користувача функцій. Для створення нової скалярною пользовательской функції в браузері об'єктів, в БД 'Bank', в папці 'Programmability', щолкнул ПКМ по папці 'Functions' і в появішемся меню вибрав пункт 'New/scalar-valued Function'. З'явиться вікно скалярною користувальницької функції.

Синтаксис столярної користувальницької функції схожий на синтаксис збереженої процедури. Однак є ряд істотних відмінностей.

- 1) Область визначення імені функції (Inline Function_Name);
- 2) Параметрів передані в процедуру (@ Param1). Визначення параметрів аналогічного визначення параметрів а збереженої процедури;

3) Тип даних значення, що повертається процедурою
 Область оголошення змінних, використовуваних всередині функції. Объявление змінних має наступний синтаксис:

4) DECLARE @ <Ім'я змінної> <Тип даних>

5) Тіло самої користувальницької функції, містить команди мови програмування запитів T-SQL;

6) Команда RETURN повертає результат виконання функції. Повинен виглядати так:

RETURN @ <Ім'я змінної з результатом>

Мінлива повинна бути того ж типу даних, яка була вказана в пункті 3.

Створю скалярную призначену для користувача функцію, яка обчислює середнє трьох величин. У вікні нової користувальницької функції набрав код представлений нижче:

```

USE [Bank]
GO
/***** Object: UserDefinedFunction [dbo].[Функция средних
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[Функция средних трех величин]
(
    @Value1 int, @Value2 int, @Value3 int
)
RETURNS Real
AS
BEGIN
    DECLARE @Result Real
    SELECT @Result = (@Value1+@Value2+@Value3)/3
    RETURN @Result
END

```

Рис.5.2-Створення функції.

Розглянувши більш детально код даної скалярної функції:

CREATE FUNCTION [Функция средних трех величин] - визначає ім'я створюваної функції як "Функция средних трех величин";

@ Value1, @ Value2, @ Value3 - визначають три параметри процедури Value0, Value2 і Value3. Даним параметрам можна привласнити цілі числа (Тип даних Int);

RETURNS Real - показує, що функція повертає дробові числа (Тип даних Real);

DECLARE @Result Real - оголошується змінна @Result для зберігання результату роботи функції, тобто дрібного числа (Тип даних Real);

SELECT @Result = (@ Value1 + @ Value2 + @ Value3) / 3 - обчислює середнє і поміщає результат в змінну @Result;

RETURN Result - повертає значення змінної @Result.

Для створення функції, виконавши вищеописаний код, натиснувши кнопку (Виконати) на панелі інструментів. У нижній дрібтячи вікна з кодом з'явиться повідомлення "Command (s) completed successfully.". Закрив вікно з кодом, клацнувши мишею по кнопці закриття розташованій у верхньому правому куті вікна з кодом функції.

Перевірив роботу створіть скалярною користувальницької функції. Для запуску користувальницької функції необхідно створити новий порожній запит, натиснувши на кнопку (Новий запит) на панелі інструментів. У вікні з порожнім

запитом набрав команду SELECT 'dbo. [Фуакція середніх трьох величин] (3, 5, 4)
Натиснув кнопку (Виконати) на копали інструментів.

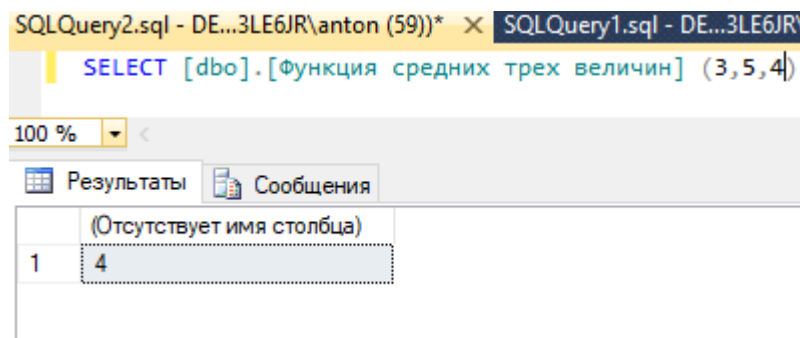


Рис.5.3-Перевірка функції.

У нижній частині вікна з кодом з'явиться результат виконання нової скалярною користувальницької функції: 4.

Тепер створю більш складну скалярную призначену для користувача функцію, призначену для визначення останнього дня місяці введеної дати.

Створив нову скалярную призначену для користувача функцію, так як про це сказано вище. У вікні нової скалярную функцію, так як про це сказано вище. у вікні користувальницької функції набрав следующий код:

```

SQLQuery3.sql - DE...3LE6JR\anton (58) X
USE [Bank]
GO
/***** Object:  UserDefinedFunction [dbo].[Последний день месяца]    Scr
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[Последний день месяца]
(
    @MyDate DateTime
)
RETURNS DateTime
AS
BEGIN
    DECLARE @Year Int
    DECLARE @Month Int
    DECLARE @Day Int
    DECLARE @TmpDate VarChar (10)
    DECLARE @Result DateTime

    SET @Year=DatePart(yy, @MyDate)
    SET @Month=DatePart(mm, @MyDate)
    SET @Day=DatePart(dd, @MyDate)

    IF @Month=12
    BEGIN
        SET @Month=1
        SET @Year=@Year-2
    END
    ELSE
    BEGIN
        SET @Month=@Month-1
    END

    SET @TmpDate=CONVERT(VarChar, @Month+'01/'+CONVERT(VarChar, @Year))
    SET @Result=CONVERT(DateTime, @TmpDate)
    RETURN @Result
END

```

Рис.5.4- скалярну призначену для користувача функцію.

Перешол до розгляду вищенаведеного коду. Код складається з наступних груп команд:

CREATE FUNCTION [Останній день місяця] - визначає ім'я створюваної функції як 'Останній день місяця';

@MyDate - визначають параметр процедури MyDate. Ця опція недоступна привласнити значення дат або часу (Тип даних DateTime);

RETURNS DatTime - показує, що функція повертає дату або час (Тип даних Data Time);

DECLARE @Year Int, DECLARE @Month Int, DECLARE @Day Int оголошуються змінні @Year, @Month і @Day для зберігання цілочисельних значень року, місяця і дня введеної дати (Тип даних Int).

DECLARE @TmpDate VarChar (10) оголошує змінну "TmpDate" для зберігання проміжного значення дати в рядку довжиною до 10 символів (Тип даних VarChar (10)).

DECLARE @Result DateTime оголошує змінну 'Result "для зберігання результату - дати останнього дня місяця (Тип даних DateTime).

SET @ Year = DatePart (yy, ®MyDate), SET @ Month = DatePart (mm, @MyDate),

SET j1Day = DatePart (dd, NMyDate) - визначаються частини введеної дати і

поміщається в змінні Year, @Month і @ Day. Для визначення частин дати

використовується функція DatePart, що має наступний синтаксис: DatePart

<частина

дати »,« дата>). Тут "частина дати" - це закодована спеціальними символами обумовлена частина дати (yy - рік, mm - місяць, dd - день), "дата" - це дата, частини якої визначаємо.

IF @ Month = 12

BEGIN

SET @ Month = 1

SET @ Year = @ Year + 1

END

ELSE

BEGIN

SET @ Month = @ Month + 1

END

Вищенаведеного фрагмент коду виконує наступні дії: Якщо номер місяці дорівнює 12 то встановити номер місяця (@ Month) рівним 1 і збільшити рік (®Year) на 1, інакше збільшити місяць на 1.

SET @ TmpDate = Convert (Varchar, @Month) + ' / 01 /' + Convert (Varchar, @Year), SET

®Result = Convert (DateTime, ®TmpDate) - переводить числові значення дати в дату в строковому форматі н записує її в змінну ®TmpDate, потім переводить дату в строковому форматі в тип даних дати і часу і поміщає її в змінну ®Result. Для конвертації використовується функція Convert, що має наступний синтаксис:

Convert (<тип даних »,« значення »), тут" тип даних 'це тип даних в який переводяться 'значення '.

SET @ Result = DateAdd (dd, -1, ®Result) - з дати, що зберігається в змінної

®Result віднімається 1 день, для цього використовується функція DateAdd, що має

наступний синтаксис:

DateAdd (<частина дати »,« кількість періодом »,« дата>) - тут <частина дати>, це закодована спеціальними символами обумовлена частина дати DatePart,

(кількість періодів) - це кількість частин дати додатків введеної датою (параметр "дата") .

REISEN @Result - повертає значення, збережене в змінної @Result.

Для створення функції, виконаємо вищеописаний код, як і у випадку з попередньої функцією, найнявши кнопку (Виконати).

Після появи повідомлення "Command (s) completed successfully." закрийте вікно з кодом.

Перевіримо роботу функції 'Останній день місяць "виконавши її. Створіть новий порожній запит, потім у вікні з порожнім запитом наберіть команду SELECT dbo. [Последній день місяця] ('12 / 07/08 ') і натиснув кнопку (Виконати) на панелі

інструментів.

З'явиться результат виконання нової скалярною користувальницької функції: 2008-12-31.

Тепер перейдемо до створення табличних призначених для користувача функцій. для

створення публічної користувальницької функції в браузері об'єктів, в БД "Bank", в папці "Programmability ", клацнув ПКМ по папці "Functions "в меню оберіть пункт "New / Table-valued Function ". З'явиться вікно нової табличній користувальницької функції.

Розглянув структуру коду табличкою користувальницької функції. Таблична призначена для користувача функція складається з наступних розділів:

Область визначення імені функції (Inline_Function_Name);

Параметри, що передаються в процедуру (@ Param1, @ Param2);

RETURNS TABLE показує що функція є табличкою, тобто повертає таблицю;

Тіло самої користувальницької функції, складається з команди SELECT мови програмування запитів T-SQL.

Решта розділів табличній призначений для користувача функції аналогічні таким самим розділам збережених процедур і скалярних пальзовательських функцій.

На закінчення розглянув створення табличкою користувальницької функції "функція відбору за віком", що обчислюють поточний вік студентів в залежності від їх дати народження. У вікні нової призначений для користувача функції набрав следующий код:

					5.151.1.42.08-ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

SQLQuery4.sql - DE...3LE6JR\anton (55))  X  SQLQuery3.sql - DE...3LE6JR\anton (58))
USE [Bank]
GO
/***** Object:  UserDefinedFunction [dbo].[Функция отбора по возрасту]
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER FUNCTION [dbo].[Функция отбора по возрасту]
(
    -- Add the parameters for the function here
)
RETURNS TABLE
AS
RETURN
    -- Add the SELECT statement with parameter references here
    SELECT FIO,[Age], Возврат = DateDiff(yy,[Age],GetDate())
    FROM dbo.[Workers1]

```

Рис.5.5- скалярную [Функция відбору за віком].

З коду представленого на рис.5.2 видно, що дана табличная функція не має параметрів і реалізується командою

SELECT ПІБ, [Дата народження], Вік = DateDiff (yy, [Дата народження], GetDate
())

FROM Студенти

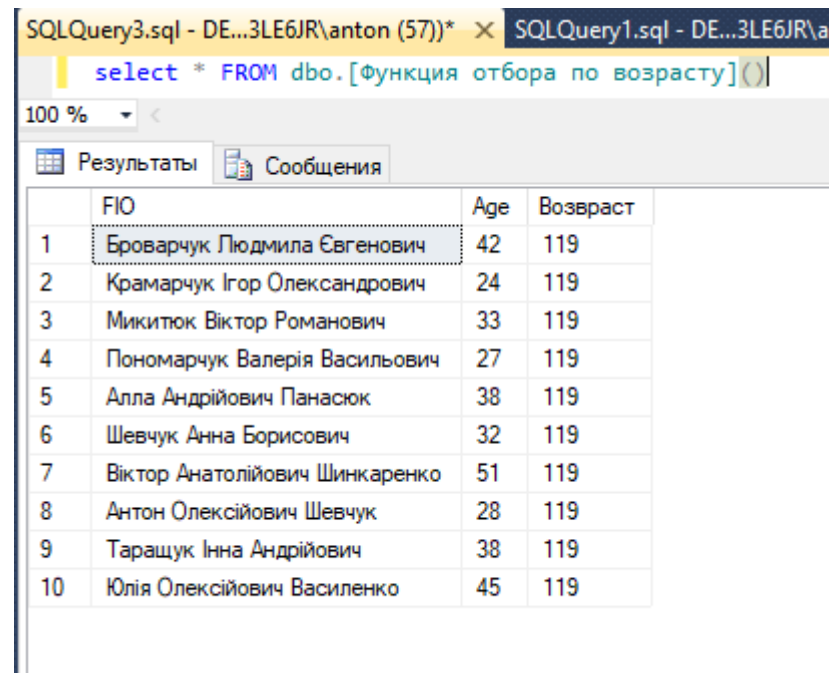
З вищепредставленими команди видно, що з таблиці "Workes1" відображаються поля "FIO" і "Age", а також обчислюваності поле "Вік" .полі "Вік" обчислюється за допомогою вбудованої функції DateDiff обчислює відмінність між датами в певних одиницях виміру (частинах дати) і має наступний синтаксис:

DateDiff (<частина дати >,< початкова дата>, <кінцева дата>).

Тут "частина дати" - це закодовані спеціальними символами одиниці виміру (частина дати) (yy - рік, mm - місяць, dd - день), "початкова дата" - дата початку періоду і "кінцева дата" - дата кінця періоду. У нашому випадку в якості початкової дати беремо дату народження студента, а в якості кінцевої дати беремо поточну дату (функція GetDate ()).

Для створення функції, виконав вищеописаний код, як і у випадку з попередньою функцією. Після появи повідомлення "Command (s) completed successfully." закриємо вікно з кодом.

Перевірив працездатність нової табличній користувальницької функції. Створив новий порожній запит, потім у вікні з порожнім запитом наберуть команду `SELECT * FROM dbo.[Функція відбору за віком] ()` і натисніть кнопку – (! Execute) на панелі інструментів (рис.5.6).



The screenshot shows a SQL query window titled 'SQLQuery1.sql - DE...3LE6JR\anton (57)*'. The query text is `select * FROM dbo.[Функція відбору за віком]()`. Below the query editor, the 'Results' tab is active, displaying a table with 10 rows and 4 columns: 'FIO', 'Age', and 'Возраст' (which is highlighted). The data is as follows:

	FIO	Age	Возраст
1	Броварчук Людмила Євгенович	42	119
2	Крамарчук Ігор Олександрович	24	119
3	Микитюк Віктор Романович	33	119
4	Пономарчук Валерія Васильович	27	119
5	Алла Андрійович Панасюк	38	119
6	Шевчук Анна Борисович	32	119
7	Віктор Анатолійович Шинкаренко	51	119
8	Антон Олексійович Шевчук	28	119
9	Тарашук Інна Андрійович	38	119
10	Юлія Олексійович Василенко	45	119

Рис.5.6-перевірка [Функція відбору за віком].

У нижній частині вікна з'явиться таблиця з прізвищами, датами народження і віком співробітника на даний момент часу (рис.6.9).

На цьому я закінчує створення призначених для користувача функції і перейшов до розгляду цілісності даних, діаграм і тригерів. Після закінчення виконання глави 2.5 оглядач об'єктів мати такий вигляд (рис.5.7):

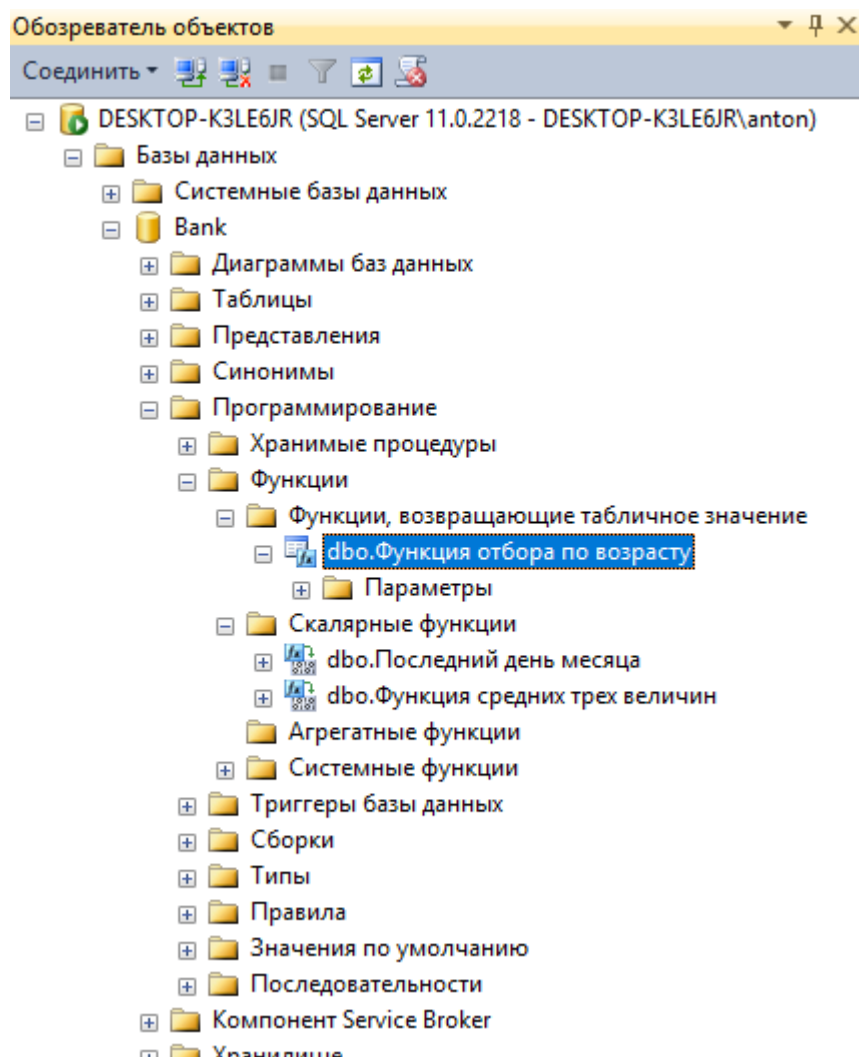


Рис.5.7-кінецьна виконеної роботи.

6. Діаграми і тригери

В БД "Microsoft SQL Server 2008" все діаграми знаходяться в папці "Database Diagrams" оглядача об'єктів (рис.6.1).

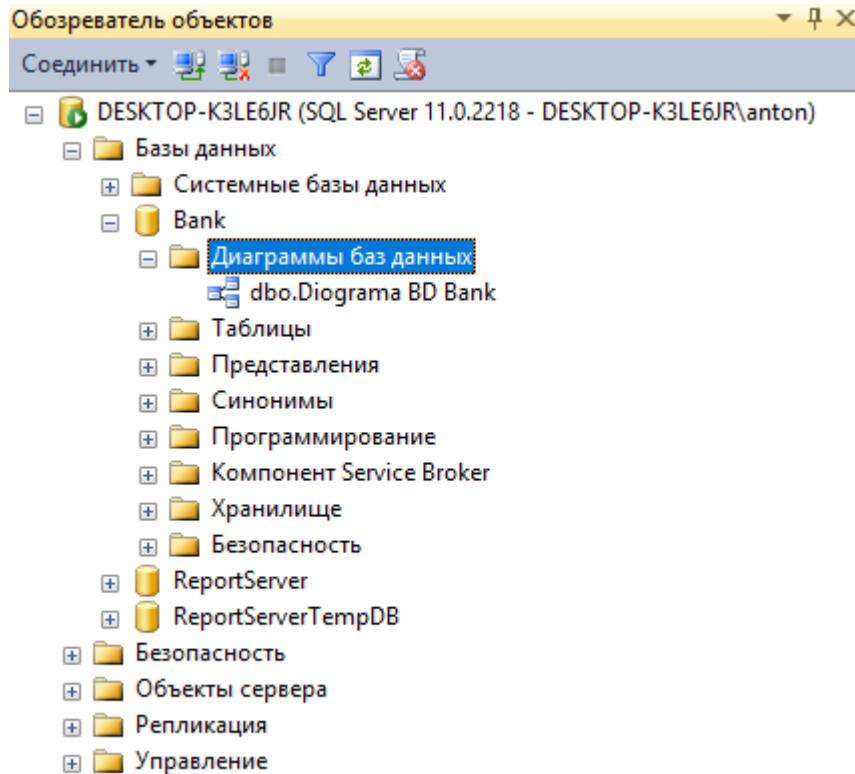


Рис.6.1-Діоргаби баз даних

Створив діаграму, що забезпечує цілісність даних БД "Bank". Щоб створити діаграму в БД "Bank" клацнув ПКМ по папці "Діограмы базы данных" і в меню вибрав пункт "Создать диограму базы данных". Спочатку з'явилася вікно з питанням про додавання нового об'єкта "Діаграма". У цьому вікні я натиснув кнопку "Да". Потім з'явиться вікно "Добавление таблицы" призначене для добовлением таблиць в нову діограму (рис.6.2).

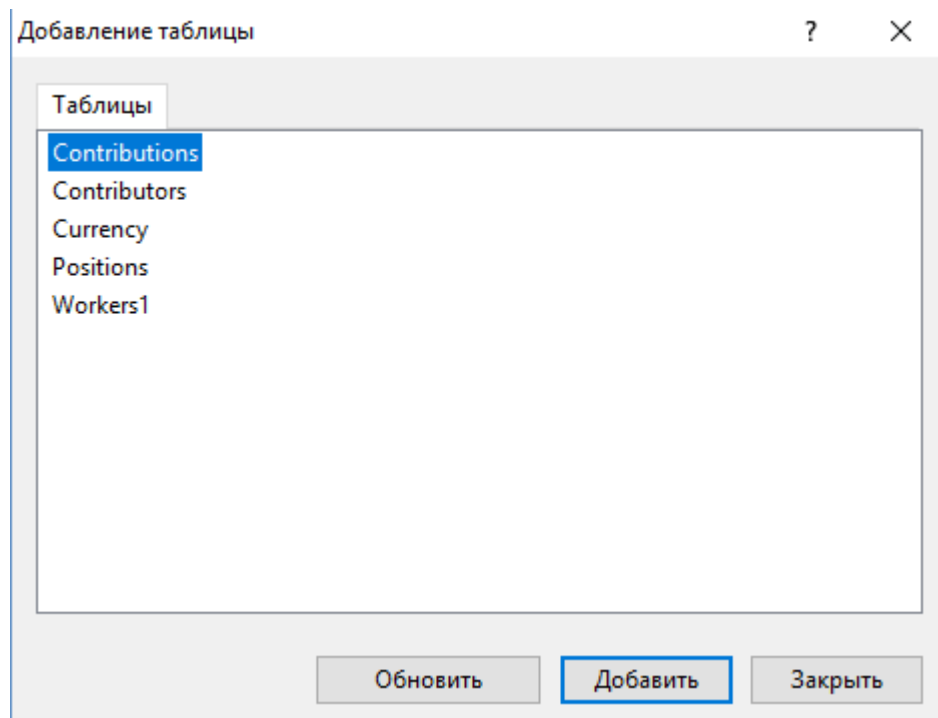


Рис.6.2-Добавлення таблиці.

У вікні додавання таблиць виділив все таблиці БД і натиснув кнопку "Додати" (рис.6.2). Закрив вікно "Діаграми бази даних" натиснув на кнопку "Закрити".

З'явиться вікно діаграми, де будуть відображені відібрані таблиці. Тепер необхідно визначити зв'язки між таблицями. Перетягнув поле " Post code " з таблиці "Workers1" на таке ж поле в таблиці "Positions".

З'явиться вікно створення зв'язку між таблицями "Таблица и столбцы" (рис.6.3).

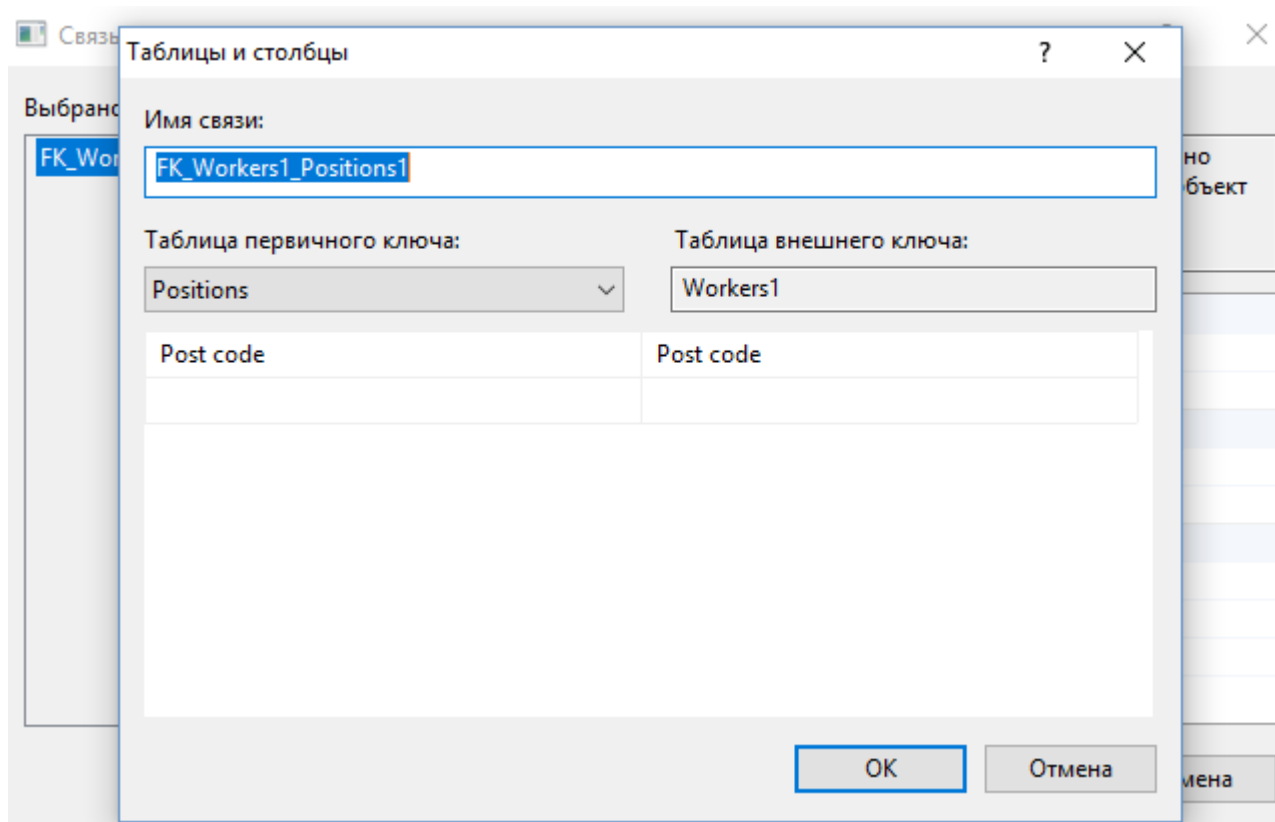


Рис.6.3-Таблиці і стовбці.

У вікні створення зв'язку натиснув кнопку "Ок". З'явиться вікно налаштування властивостей зв'язку "Связь по внешнему ключу".

Оставил свойства связи без изменений и в окне свойств связи нажмие кнопку "Ок". В диаграмме между таблицами "Workers1" и "Positions" появится связь в виде ломанной линии (рис.6.4).

Аналогічним чином зв'язок таблиці "Workers1" з таблицею "Contributors", перетягнувши поле "Employee code" з таблиці "Workers1" на однойменне поле в таблиці "Contributors". Потім, зв'язав таблиці "Contributors" і "Contributions", перетягнувши поле "Deposit code". Потім, зв'язав таблиці "Contributions" і "Currency", перетягнувши поле "Currency code". Після виконання перерахованих вище дій діаграма прийме такий вигляд (рис.6.4).

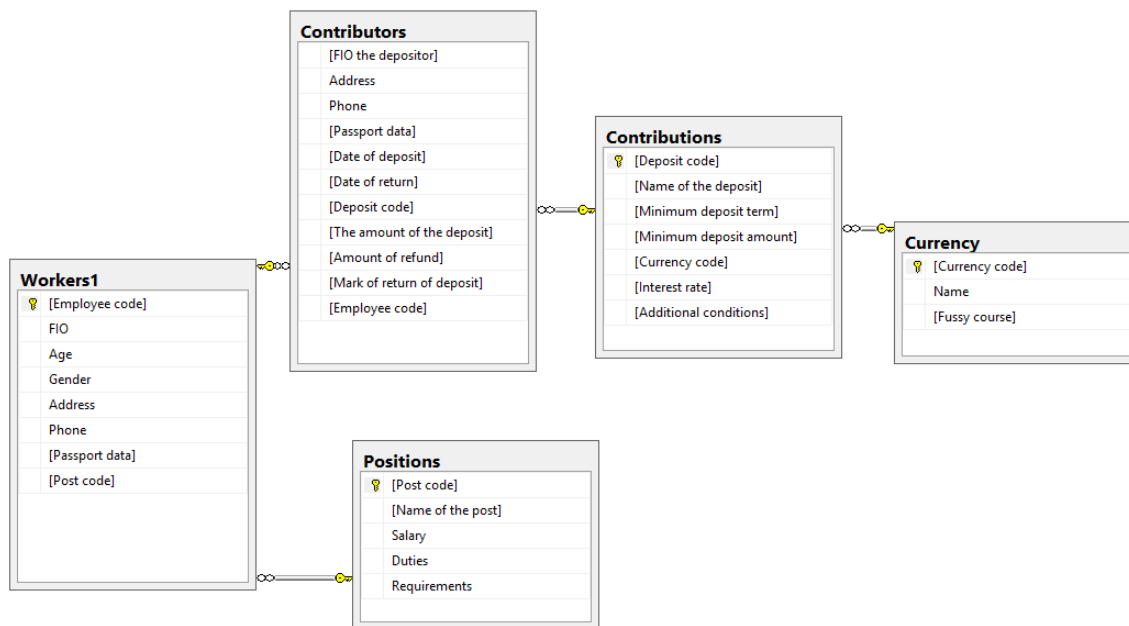


Рис.6.4-зв'язки таблиц в діаграмі.

Закрив вікно з діаграмою, клацнувши мишею по кнопці закриття розташованої у верхньому правому куті вікна з діаграмою. З'явиться вікно з питанням про збереження нової діаграми, де необхідно натиснути кнопку "Так".

З'явилося вікно визначення імені нової діаграми. У вікні визначення імені, задайте ім'я діаграми як "Diograma BD Bank" і натиснув кнопку "ОК".

Перешол до створення тригерів. Створив тригери для таблиці "Workers1". Тригери створюються окремо для кожної таблиці і розташовуються в браузері об'єктів у програмі "Триггеры". У моєму випадку, папка "Триггеры " входить до складу таблиці "Workers1".

Для початку створимо тригер, що виводить повідомлення "Запис додана" при додаванні запису в таблицю "Workers1". Создал новий тригер, клацнувши ПКМ по папці "Триггеры" в таблиці "Workers1" і в меню вибравши пункт "Создать тригер".


```

SQLQuery4.sql - DE...3LE6JR\anton (59)) X
USE [Bank]
GO
/***** Object: Trigger [dbo].[Индикатор добавления]    Script Dat
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER TRIGGER [dbo].[Индикатор добавления]
ON [dbo].[Workers1]
AFTER INSERT
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for trigger here
PRINT 'Запись добавлена'

END

```

Рис.6.5-створення тригерів.

З рис.6.5 видно, що створюваний тригер "Индикатор добавления" виконується після додавання запису (AFTER INSERT) в таблицю "Workers1" (ON dbo.Workers1). Після додавання запису тригер виведе на екран повідомлення "Запис додана" (PRINT 'Запис додана'). Виконав набраний код, натиснувши кнопку Execute на панелі інструментів. У нижній частині вікна з кодом з'явиться повідомлення "Command (s) completed successfully."

Перевірив, як працює новий тригер. Створіть новий порожній запит і в ньому наберіть наступну команду для додавання нового запису в таблицю "Студенти" (рис.6.6):

```

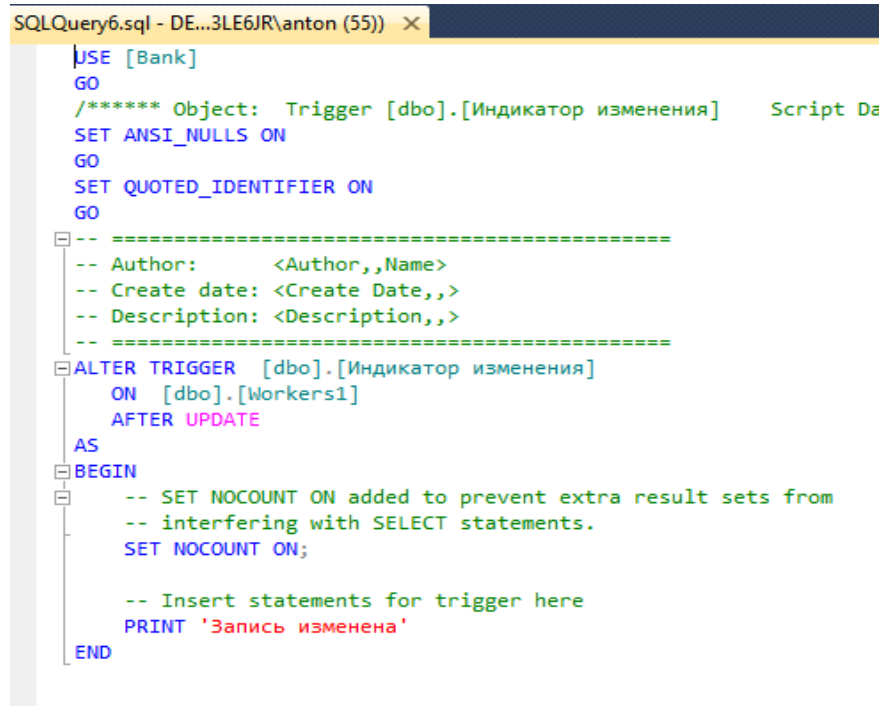
SQLQuery5.sql - DE...3LE6JR\anton (55))* X  DESKTOP-K3LE6JR.Bank - dbo.Workers1  SQLQue
INSERT INTO dbo.Workers1
Values
'Юлія|Олексійович Василенко'
<'45'
<'Жін'
<'07520, Львівська область, місто Львів, просп. Генерала Жадова, 87'
<'+380671640544'
<'МД-3567646'
<'З'

```

Рис.6.6-створення тригерів "Студенти".

Виконав набрану команду, натиснувши кнопку Execute на панелі інструментів. У таблицю буде доданий новий запис, і тригер виведе повідомлення "Запис додана" (рис.6.6).

Тепер створимо тригер відображає повідомлення "Запис змінена". Створив новий тригер, як в попередньому випадку. У вікні нового тригера набереть наступний код (рис.6.7):



```

SQLQuery6.sql - DE...3LE6JR\anton (55)
USE [Bank]
GO
/***** Object: Trigger [dbo].[Индикатор изменения]    Script Date: 10/10/2015 11:58:11 AM
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER TRIGGER [dbo].[Индикатор изменения]
ON [dbo].[Workers1]
AFTER UPDATE
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for trigger here
PRINT 'Запись изменена'
END

```

Рис.6.7- тригер "Індикатор зміни"

З (рис.6.7) видно, що новий тригер "Індикатор зміни" виконується після зміни запису (AFTER UPDATE) в таблиці "Workers1" (ON dbo.Workers1). Після зміни записи тригер виведе на екран повідомлення "Запис змінена" (PRINT 'Запис змінена'). Виконайте набраний код. У нижній частині вікна з кодом з'явиться повідомлення "Command (s) completed successfully.".

Перевірив працездатність створеного тригера. Створив новий запит і в ньому наберіть команду, представлену на (рис.6.8).

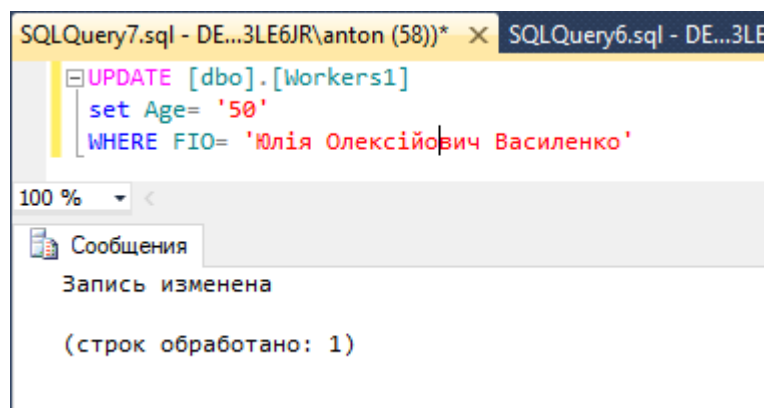


Рис.6.8- Виконав набрану команду.

Виконав набрану команду, натиснувши кнопку на панелі інструментів. У таблицю буде доданий новий запис, і тригер виведе повідомлення "Запис змінена" (рис.6.8).

Для повноти картини створив тригер, що виводить повідомлення при видаленні запису з таблиці "Workers1". Створіть новий тригер і в ньому наберіть код, показаний на (рис.6.9).

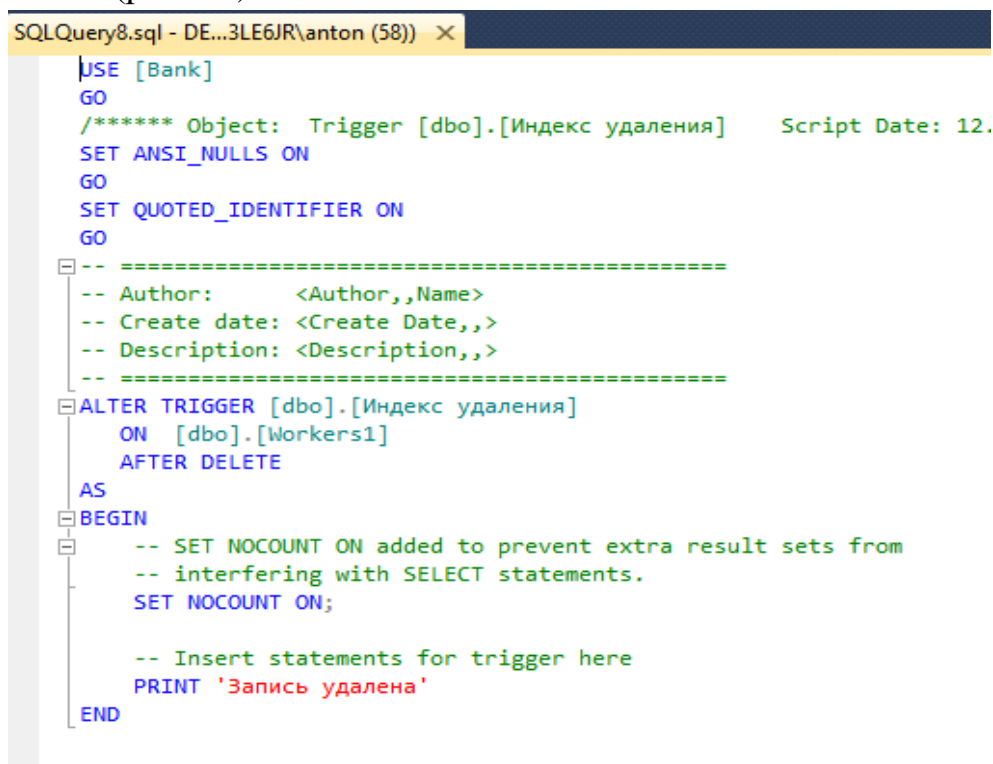


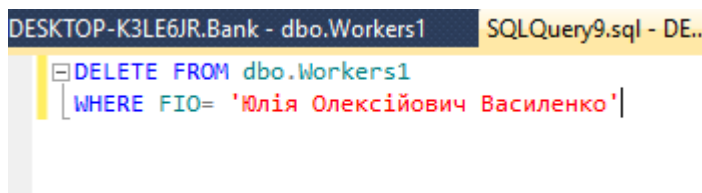
Рис.6.9- Створюваний тригер "Індикатор видалення".

Створюваний тригер "Індикатор видалення" виконується після видалення запису (AFTER DELETE) з таблиці студенти (ON dbo.Workers1). Після видалення

запису триггер виводить повідомлення "Запис видалена" (PRINT 'Запис вилучена').

Виконайте код, представлений (рис.6.9). У нижній частині вікна з кодом з'явиться повідомлення "Command (s) completed successfully."

Перевіримо роботу тригера "Індикатор видалення" видаливши створену раніше запит з таблиці "Workers1". Для цього створіть новий запит і в ньому наберіть і існуючу команду (рис.6.10):



```
DESKTOP-K3LE6JR.Bank - dbo.Workers1  SQLQuery9.sql - DE..  
DELETE FROM dbo.Workers1  
WHERE FIO= 'Юлія Олексійович Василенко'
```

Рис.6.10- Виконав вищенаведену команду.

Виконав вищенаведену команду. Після видалення запису триггер "Індикатор видалення" відобразить повідомлення "Запис видалена" (рис.6.10).

На закінчення розглянув приклад застосування тригерів для забезпечення цілісності даних. Створив тригер "Видалення студента", який при видаленні запису з таблиці студенти спочатку видаляє всі пов'язані з нею записи з таблиці "Оцінки", а потім видаляє саму запис з таблиці "Студенти", тим самим забезпечується цілісність даних.

Створив новий тригер і в ньому наберать наступний код (рис.6.11):

```

SQLQuery10.sql - D...3LE6JR\anton (57)) X
USE [Bank]
GO
/***** Object: Trigger [dbo].[Удаление студентов]    Script Date
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER TRIGGER [dbo].[Удаление студентов]
ON [dbo].[Workers1]
INSTEAD OF DELETE
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for trigger here
DELETE dbo.Contributors
FROM Deleted
WHERE Deleted.[Employee code]=Contributors.[Employee code]
DELETE dbo.Workers1
FROM Deleted
WHERE Deleted.[Employee code]=Workers1.[Employee code]
END

```

Рис.6.11- Створюваний тригер "Видалення студента".

Створюваний тригер "Видалення студента" виконується замість видалення запису (INSTEAD OF DELETE) з таблиці "Workers1" (ON dbo.Workers1).

При спрацьовуванні тригера замість видалення запису створюється тимчасова константа Deleted, що містить ім'я таблиці з якої мало бути процес ведено видалення.

Після спрацьовування тригера з таблиці "Contributors" видаляється запис, у якій пов'язані з поля "Employee code" дорівнює значенню такого ж поля у видаляється записи з таблиці "Workers1". Цю операцію виконують такі команди:

```
DELETE FROM dbo.Contributors
```

```
WHERE Deleted. [Employee code] = Contributors. [Employee code]
```

Навіщо видаляється запис з таблиці "Workers1", яку видаляли до спрацьовування тригера. Видалення виконується наступними командами:

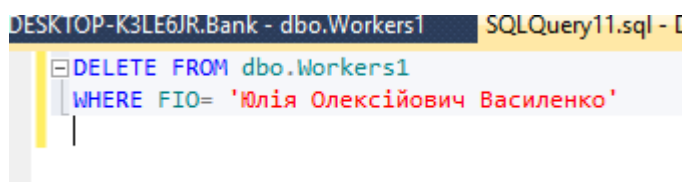
```
DELETE dbo.Workers1
```

```
FROM Deleted
```

WHERE Deleted. [Employee code] = Workers1. [Employee code]

Виконайте код, представлений на (рис.6.11). У нижній частині вікна з кодом з'явиться повідомлення "Command (s) completed successfully."

Перевірів, як працює тригер "Видалення студента". Для цього створіть новий тригер в ньому наберіть наступний код (рис.6.12):



```
DESKTOP-K3LE6JR.Bank - dbo.Workers1 SQLQuery11.sql - I
DELETE FROM dbo.Workers1
WHERE FIO= 'Юлія Олексійович Василенко'
```

Рис.6.12- Виконайте код.

При спрацьовуванні тригера спочатку з таблиці "Contributors" втечуть всі пов'язані з видаленням записом записи, а потім видалення сама видалення запис з таблиці "Workers1", при цьому зберігається цілісність даних. На цьому я завершив роботу з діаграмами і тригерами.

7. Створення проекту. Підключення файлу даних до проекту

Тепер розглянемо створення призначеного для користувача інтерфейсу нашої БД в "Microsoft Visual Studio 2008".

Перш ніж створювати новий проект в "Microsoft Visual Studio 2008", необхідно запустити "SQL Server Browser" (Оглядач SQL серверів) сервіс, що забезпечує доступ сторонніх додатків до "Microsoft SQL Server 2008". Для Запуску чанного сервісу необхідно запустити утиліту "SQL Server Configuration Manager" (Менеджер налаштувань SQL сервера), що входить до складу "Microsoft SQL Server 2008". Для запуску даної утиліти в меню "Пуск" виберіть пункт "программа / Microsoft SQL Server 2008 / Configuration Tools / SQL Server Configuration Manger".

У правій частині вікна менеджера налаштувань SQL сервера, в списку сервісів "Microsoft SQL Server 2008" виділіть сервіс "SQL Server Browser", потім на панелі інструментів натисніть кнопку "Start Service" (Запустити сервіс). Після запуску вищевказаного сервісу вікно менеджера налаштувань SQL сервера можна закрити.

Потім перейшов безпосередньо до створення призначеного для користувача інтерфейсу нашої БД "Bank" в "Microsoft Visual Studio 2008". Для початку необхідно створити новий проект. Для цього запустіть "Microsoft Visual Studio 2008". вибравши в меню "Пуск" пункт "Програми / Microsoft Visual Studio 2008 / Microsoft Visual Studio 2008".

З'явилося вікно зі стартовою сторінкою "Microsoft Visual Studio 2008 (Start Page)" (рис.7.0).

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

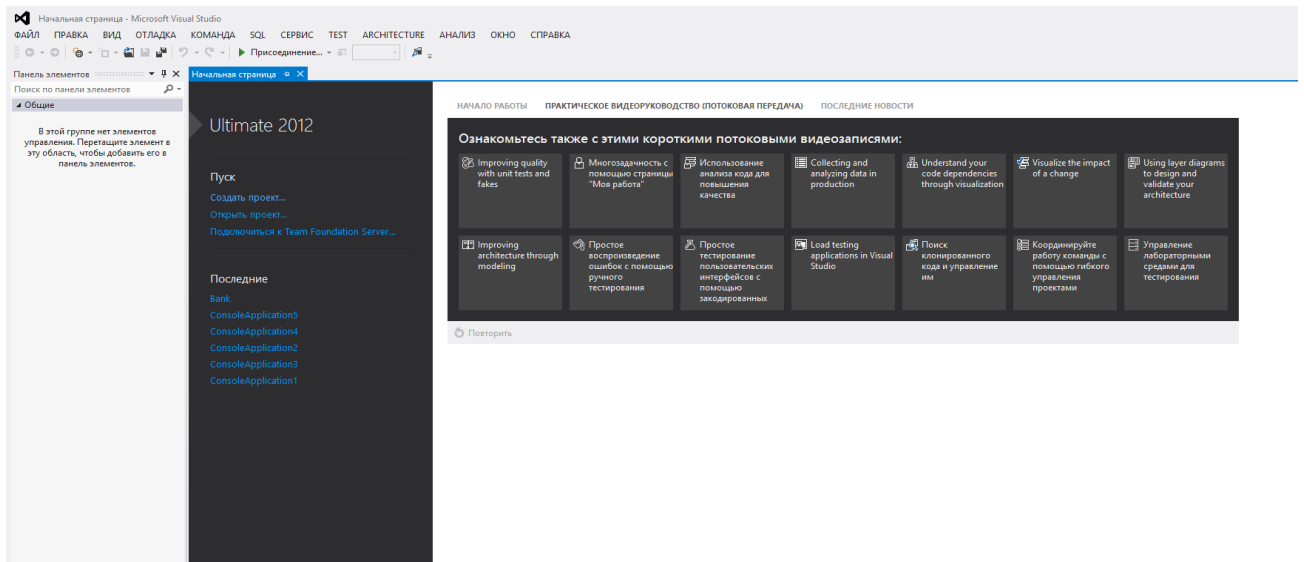


Рис.7.0- «Microsoft Visual Studio 2012»

Для створення нового проекту на стартовій сторінці в області "Recent Projects" (Недавні проекти) необхідно клацнути ЛКМ по посиланню "Create: Project ..."

(Створити: проект ...) (рис. 7.0). З'явиться вікно вибору типу створюваного проекту, і використовуваної мови програмування "New Project" (Новий проект) (рис.7.1).

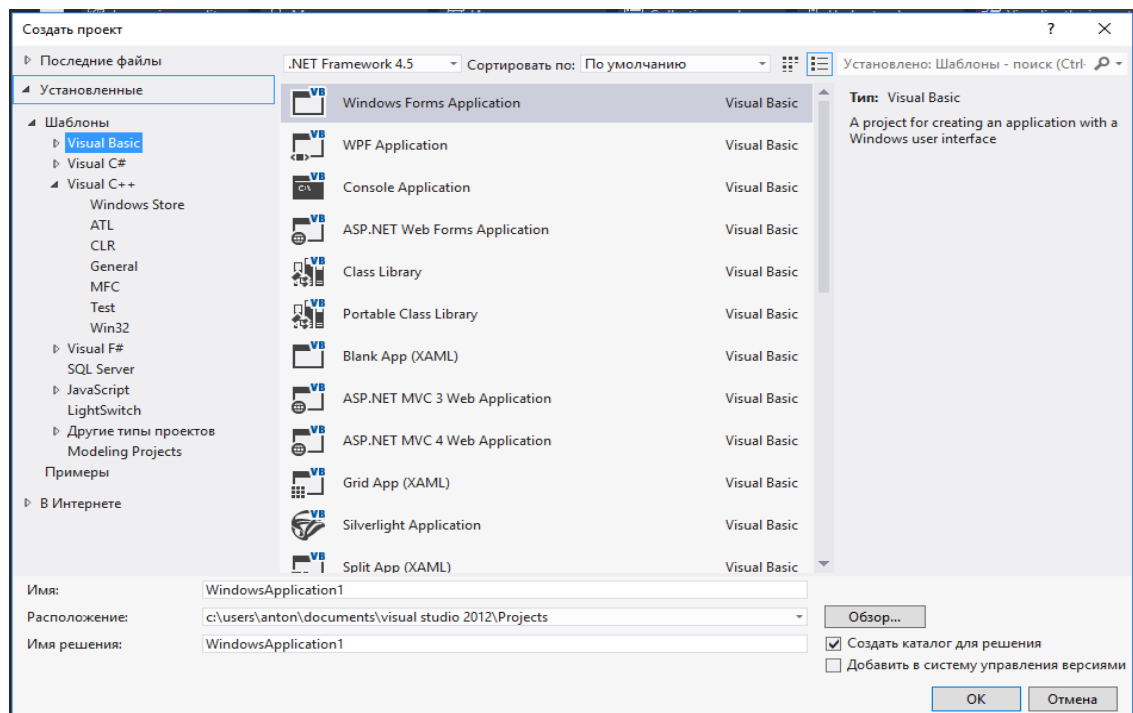


Рис.7.1- Створення проекту.

У нашому випадку на дереві типів проекту "Project types:" (Типи проектів) виберіть "Visual Basic \ Windows", а як шаблон проекту (Область Templates :)

виберіть "Windows Forms Application" (Додаток Windows). Як ім'я проекту (Поле введення Name :) задайте "Bank" і натисніть кнопку "Ок" (рис.7.1).

З'явиться стандартне вікно розробки мови "Microsoft Visual Basic 2008" (рис.7.2).

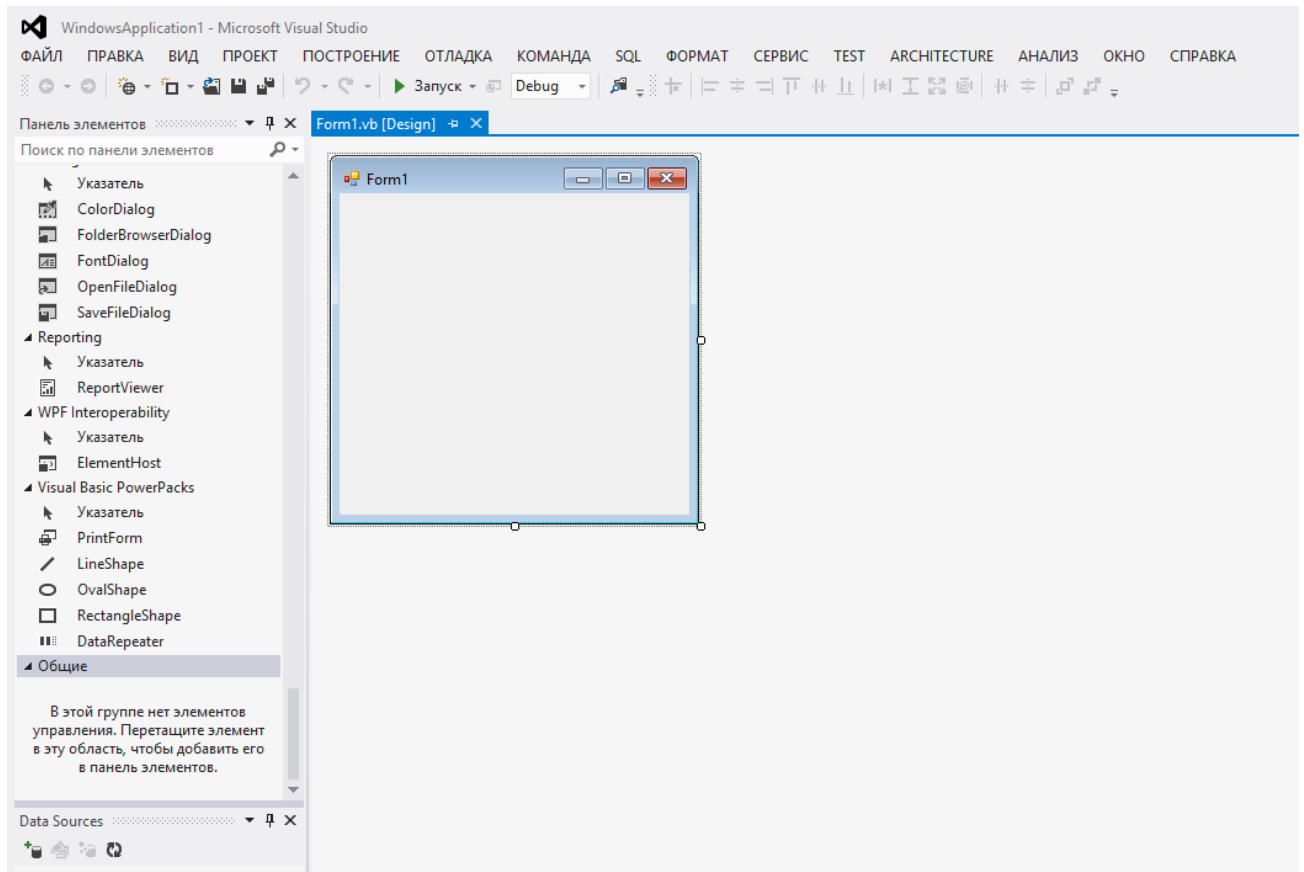


Рис.7.2-Створення форми.

Вікно розробки мови "Microsoft Visual Basic 2008" має наступну.

1. Віконне меню - містить повний набір команд для управління середовищем розробки;
2. Панель інструментів - містить кнопки з найбільш часто використовуваними командами середовища розробки;
3. Панель об'єктів (Toolbox) - містить кнопки класів для створення різних об'єктів ('Елементів управління');

4. Оглядач проекту / Джерела даних (Solution Explorer / Data Sources) до залежності активізованою в нижній частині панелі вкладки, відображає оглядач, проекту або джерела даних, підключені до проекту. Дивляться. проекту (Solution Explorer) відображає всі файли, що входять в проект і дозволяє переключатися між ними. Джерела даних це бази даних, служби або об'єкти з яких надходять дані в проект;

5. Панель властивостей (Properties) - відображає і дозволяє змінювати властивості обраного про об'єкта;

6. Робоча область - в залежності від обраної вкладки, розташованої у верхній частині області, відображає область дизайну форми, код форми або стартову сторінку.

Поснов створення нового проекту необхідно підключити до проекту створену раніше "Microsoft SQL Server 2008" БД "Students". Для підключення БД до отвору ввіконному меню середовища розробки виберіть пункт "Data \ Add New Data Source".

З'явиться вікно майстра підключення до нового джерела даних "Data Source Configuration Wizard" (рис.7.3).

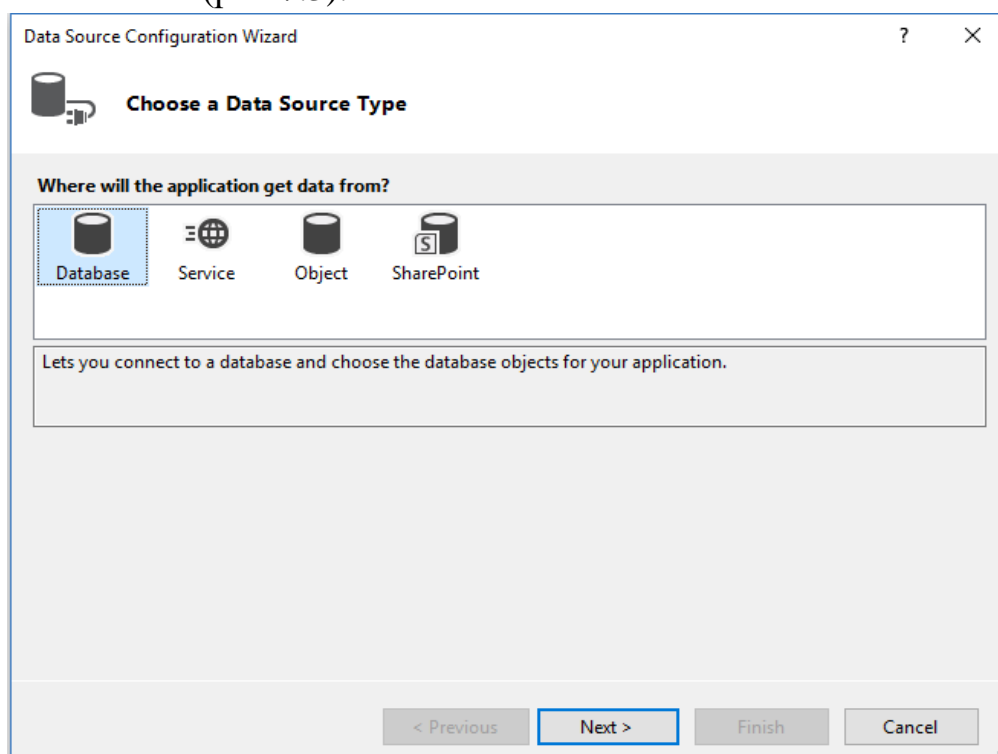


Рис.7.3- Вікно налаштування джерела даних.

У вікні вибору підключення до БД, для створення нового підключення натисніть кнопку "New Connection ...". З'явиться вікно додавання нового з'єднання "Add Connection" (рис. 7.4).

Рис.7.4- вікно додання нового з'єднання.

У вікні "Add Connection" в випадаючому списку "Server Name" (Ім'я сервера) вибрав ім'я сервера заданий при установці SQL сервера. В якості логіна і пароля для входу на сервер (Log on to the server) виберіть "Use- Windows Authenlication" (Використовувати логін і пароль облікового запису Windows). Як БД для підключення (Connect to a database) зі списку "Select or enter a database name:" (Виберіть або введіть ім'я БД) виберіть БД "Bank" (рис. 7.4).

Для перевірки працездатності створюваного з'єднання натисніть кнопку "Перевірити підключення". З'явиться повідомлення "Перевірка підключення виконано", що говорить про те, що з'єднання працездатний.

Закрив вікно, а потім у вікні додавання новою з'єднання "Add Connection" (рис. 7.4) натисніть кнопку "Ok". Сталося б повернення до вікна вибору підключення

до БД (Choose Your Data Connection). Перегляньте створену рядок підключення "Connection string", клацнувши по знаку "+" в нижній частині вікна (рис.7.5).

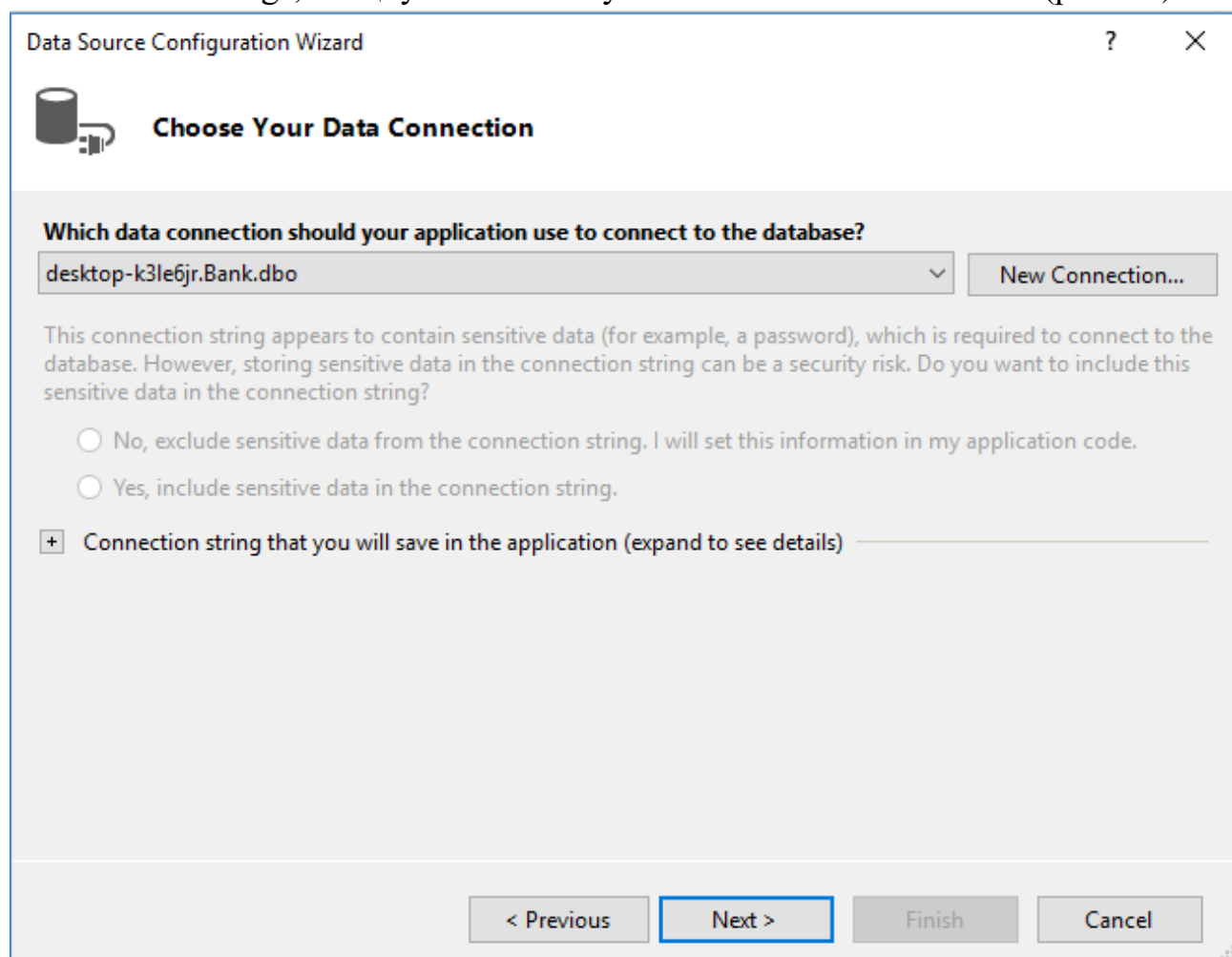


Рис.7.5- вікно підключення БД після налаштувань.

У вікні вибору підключення до БД (Choose Your Data Connection) натиснув кнопку "Next" (Далі). З'явиться вікно із запитом про збереження рядка підключення "Save the Connection String to the Application Configuration File" (Зберегти рядок підключення в файл установки програми) (рис. 7.5).

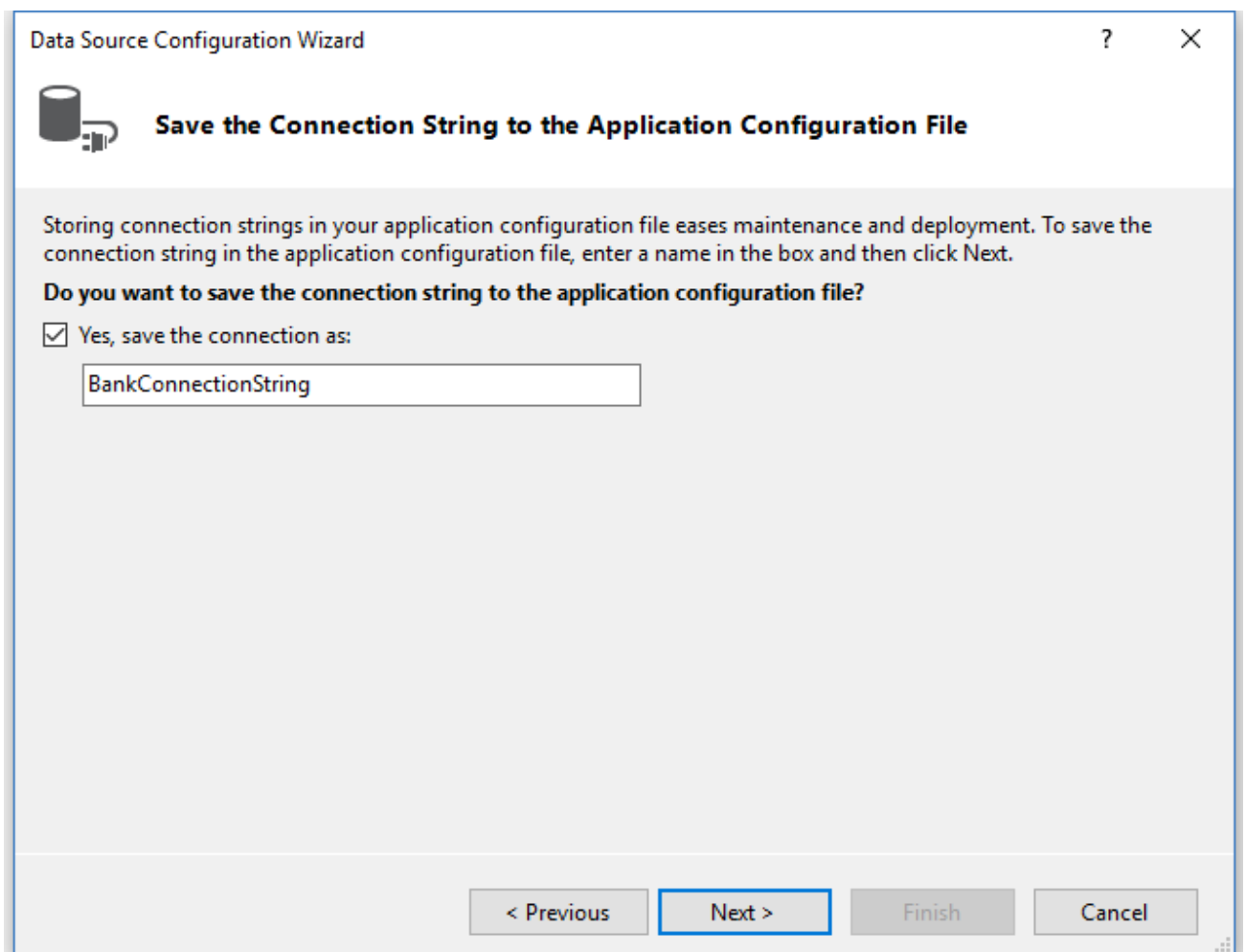


Рис.7.6- вікно вибору ім'я з'їднання.

Для збереження рядка підключення включите опцію "Yes, save the connection as: " (Так, зберегти підключення як :) і натисніть кнопку" Next ".

З'явиться вікно вибору об'єктів підключається БД (Choose Your Database Objects) (рис. 7.6).

Виберіть всі об'єкти як це показано на (рис.7.6) і натисніть кнопку "Finish" (Готово). Підключення завершено.

Для перегляду джерела даних клацніть по вкладці "Data Sources" у вікні розробки мови "Microsoft Visual Basic 2008" (рис.7.2). Вкладка "Data Sourcer".

В даному вікні в полі введення "Name:" (Ім'я) задайте ім'я проекту "Bank". В поле "Location:" (Розташування :) задайте папку, де буде збережений новий

проект, натиснувши кнопку "Browse ..." (Вибрати). Після визначення імені проекту і його місцеположення натисніть кнопку "Save" (Зберегти).

8.Створення головною кнопкової форми. Створення простих стрічкових форм для роботи з даними

Створення інтерфейса користувача. Створімо головну кнопкову форму. Для цього відкрийте Останній проект 'БД Bank '.

Після з'явилися стандартного вікна середовища розробки на робочій області додали надпис'Label 'та кнопку'Button' як показано на рис.8.1.

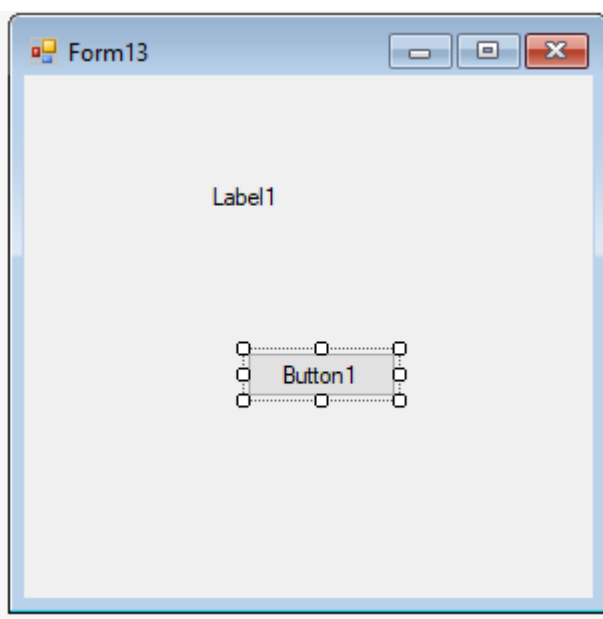


Рис.8.1-створення головної форми.

Потім заповнив Label 1 та дамо ім'я Button.Для цього в оглядачі вирішень задаємо 'Text' для 'Button' та 'Label 1'. Готовий вигляд титульної сторінки буде виглядати як на (рис.8.5).

Далі створимо головну сторінку з усіма таблицями та посилання з титульної сторінки. Для цього потрібно на розділ 'Проект'-'Додати форму Windows...'- 'Windows Forms'.

Після відкриття нової форми додав на неї Label 1 а також п'ять Button. Назвав їх у відповідності 'Працівники','Посади','Вклвди','Валюта','Додаткові 'Вкладники'.

Тепер створимо посилання с першої форми (Титульна сторінка) на другу (головну сторінку). Для цього на першій формі двічі натисніть на Button1 та впишіть наступний код (рис.8.2).

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Form3.Show()
End Sub
```

Рис.8.2-перехід на іншу Form3.

Потім перейшов до таблиць в формах. Спочатку створимо Form3 для таблиці ‘Працівники’ та створимо посилання на неї з головної сторінки. Це можна зробити, клацнувши двічі на кнопку ‘Працівники’ та вписавши код (рис.8.3).

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Form2.Show()
End Sub
```

Рис.8.3-перехід на іншу Form2.

На порожню форму додайте Label 1.Далі в ‘Джерела даних’ відкрийте таблицю ‘Prositons’ та винесіть ті дані, які вам потрібні.

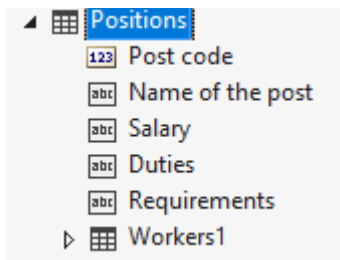


Рис.8.4- відкрив таблицю ‘Prositons’.

Сторінкова форма вкінці буде виглядати як:

Рис.8.5-Form2 Процівники.

По анології зробиа настубні форми.

Form4

Валюта

Currency code:

Name:

Fussy course:

Добавить

Сохранить

Удалить

Таблица

Рис.8.6- Таблица Валюта

Form5

Посади

Добавить

Post code:

Name of the post:

Salary:

Duties:

Requirements:

Сохранить

Удалить

Таблица

⏮ ⏪ ⏩ ⏭

Рис.8.7-Таблица Посади:

Form6

Вклады

Deposit code:

Name of the deposit:

Minimum deposit term:

Minimum deposit amount:

Currency code:

Interest rate:

Additional conditions:

Добавить

Сохранить

Удалить

Таблица

⏮ ⏪ ⏩ ⏭

Рис.8.8- Таблица Вклады

Form7

Вкладники

Добавить

Сохранить

Удалить

Таблица

FIO the depositor:

Address:

Phone: +38() - - -

Passport data: ЕК-

Date of deposit: 18 мая 2019 г.

Date of return: 18 мая 2019 г.

Deposit code:

The amount of the deposit:

Amount of refund:

Mark of return of deposit:

Employee code:

Navigation buttons:

Рис.8.9- Таблица Вкладники

На цьому я закінчив створення простих стрічкових та головної кнопочної форми.

9. Створення складних стрічкових форм для роботи з даними

Модернізація форми для таблиці "Працівники". Спочатку продублював кнопку панелі навігації, розположену в верхній часті сторінки. Додав 7 кнопок(Button) на форму нашої таблиці та перейменуйте їх як показано на (рис.9.1).

Рис.9.1- Модернізація форми для таблиці "Працівники".

Двічі клацніть ЛКМ по кнопці "Попередня" і у вікні коду форми "Form3" в процедурі "Button2Click" наберіть команду для переходу до попереднього запису:

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Workers1BindingSource.MovePrevious()
End Sub
```

Двічі клацніть ЛКМ по кнопці "Перша" і у вікні коду форми "Form3" в процедурі "Button1Click" наберіть команду для переходу до першого запису:

```
Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    Workers1BindingSource.MoveFirst()
End Sub
```

Двічі клацніть ЛКМ по кнопці "Додати" і у вікні коду форми "Form3" в процедурі "Button3Click" наберіть команду для додавання нового запису:

```
Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
    Workers1BindingSource.AddNew()
End Sub
```

Двічі клацніть ЛКМ по кнопці "Остання" і у вікні коду форми "Form3" в процедурі "Button4Click" наберіть команду для переходу до останнього запису:

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    Workers1BindingSource.MoveLast()
End Sub
```

Двічі клацніть ЛКМ по кнопці "Наступна" і у вікні коду форми "Form3" в процедурі "Button5Click" наберіть команду для переходу до наступного запису:

```
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    Workers1BindingSource.MoveNext()
End Sub
```

Двічі клацніть ЛКМ по кнопці "Удалить" і у вікні коду форми "Form3" в процедурі "Button6Click" наберіть команду для видалення запису:

```
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    Workers1BindingSource.RemoveCurrent()
End Sub
```

End Sub

Двічі клацніть ЛКМ по кнопці "Сохранить" і у вікні коду форми "Form3" в процедурі "Button7Click" наберіть команду для сохранения запису:

```
Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
    Me.Validate()
    Me.Workers1BindingSource.EndEdit()
    Me.TableAdapterManager.UpdateAll(Me.BankDataSet)
End Sub
```

Аналогічно, зробіть інші таблиці.

Тепер змінимо об'єкти, які відображають поля для більш комфортного вводу інформації. Для цього в відповідній формі потрібно видалити TextBox таких полів, як : 'стать', 'телефон', 'паспортні дані'

Для відображення ‘Телефон’ та ‘Паспортні дані’ будемо використовувати текстові поля вводу по масці(MaskedTextBox). Створимо поля праворуч від надписів. Тепер на стрілочці на полі натиснемо ‘Додати маску’. З’явиться вікно для вибору масок(Рис.9.2). Для поля ‘Код співробітника’ обираємо маску ‘Numeric(5-digits)’, а для паспортних даних і телефону задайте маску <Специальный>.

Маска ввода

Выберите готовое описание маски из списка ниже или задайте специальную маску, выбрав команду "Специальный".

Описание маски	Формат данных	Проверка типа
Phone number	(574) 555-0123	(нет)
Phone number no area co...	555-0123	(нет)
Short date	12/11/2003	DateTime
Short date and time (US)	12/11/2003 11:20	DateTime
Social security number	000-00-1234	(нет)
Time (European/Military)	23:20	DateTime
Time (US)	11:20	DateTime
Zip Code	98052-6399	(нет)
<Специальный>		(нет)

Маска: €Д-0000000

Предварительный просмотр: €Д-_____

☒ Использовать ValidatingType

OK Отмена

Рис.9.2-створення маски.

Тепер необхідно під'єднати створені текстові поля вводу по масці к відповідним полям. Для цього з вкладки 'Источники данных' перетяніть відповідні поля на текстове поле вводу по масці ,які знаходяться праворуч від надписів.

Відобразимо поле 'Телефон'. Для цього на вкладці 'Источники данных', праворуч від стовбця 'Телефон' таблиці 'Працівники' клацніть кнопку та виберіть 'ComboBox'. Перетягніть цей стовбець у форму на відповідне місце.

На цьому ми завершуємо стрічкові форми та переходимо до табличних форм.

10. Створення табличних форм

Розглянемо створення табличної форми на прикладі форми, яка буде відображати таблицю ‘Працівники’. Для цього створимо нову форму та помістимо на неї такі об’єкти :

- 3 надписи(Label)
- 4 кнопки(Button)
- випадаючий список(ComboBox)
- текстове поле вводу(TextBox)
- групуючу рамку(GroupBox)
- список(ListBox)
- два перемикача(RadioButton)

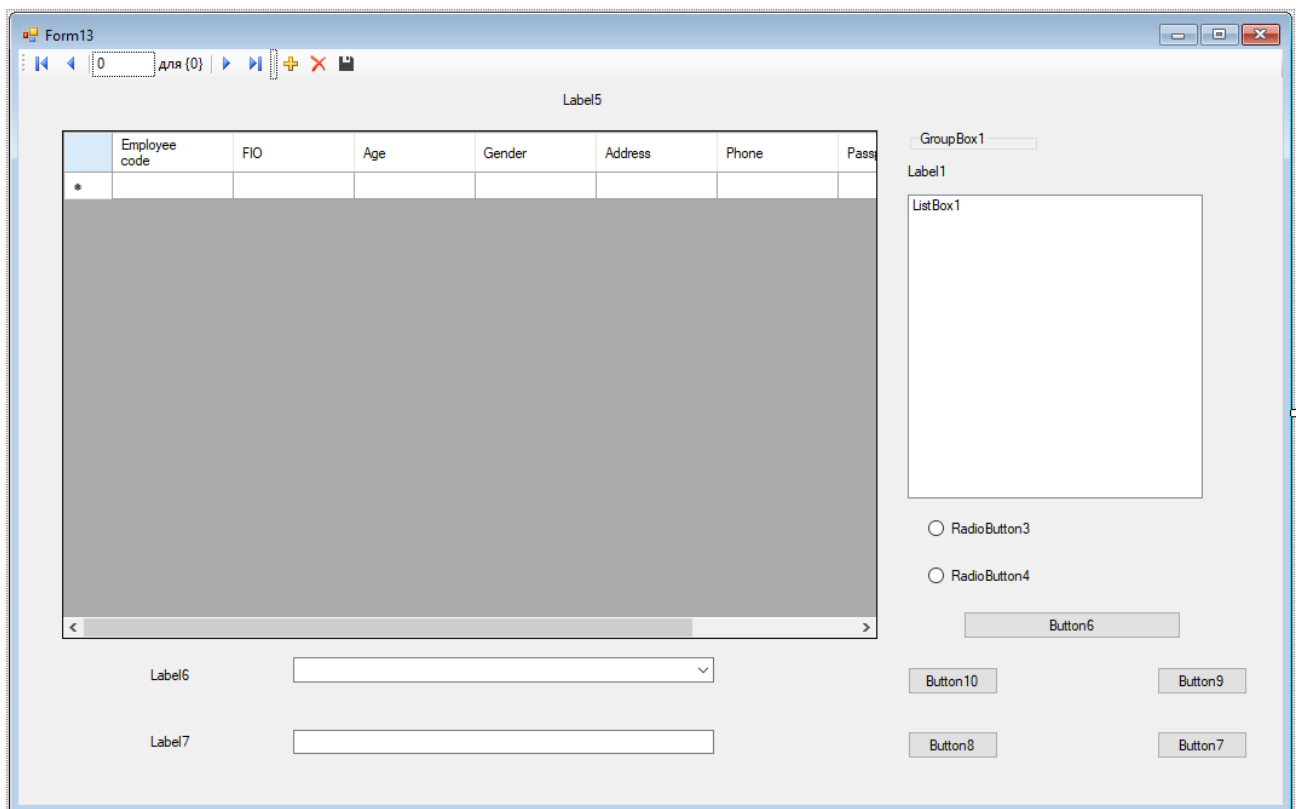


Рис.10.1- створення табличної форми.

Тепер змінимо ім’я об’єктів. Кнопки 1,2,3,4,5 як ‘Сортувати’, ‘Фільтрувати’, ‘Показати всі’, і ‘Закрити’ відповідно. Групуюча рамка – ‘Сортування’, перемикачі – ‘Сортування по зростанню’(також задати Checked = True),

‘Сортування по спаданню’. Також заповніть список значеннями з таблиці (Рис.10.2)

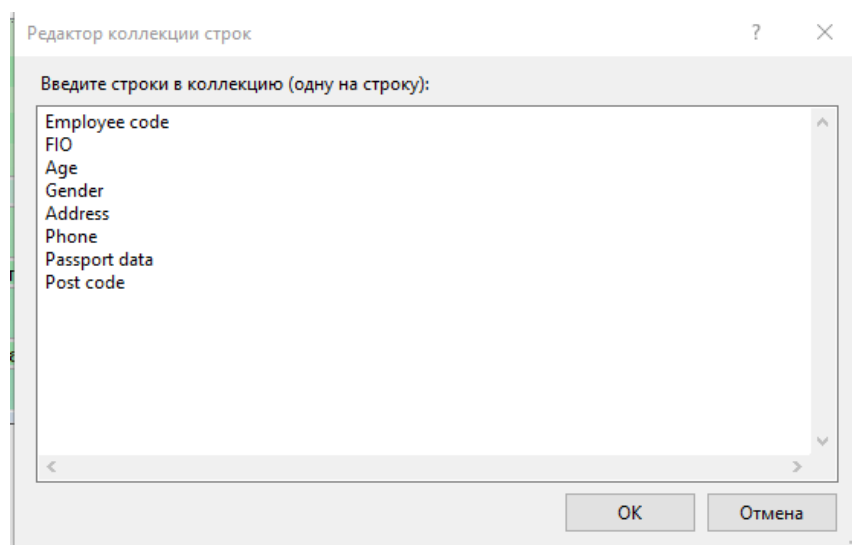


Рис.10.2- Змінимо ім'я об'єктів.

Налаштував заповнення випадаючого списку іменами співробітників з таблиці сотрудніуі. відобразив меню дій випадаючого списку. Включив опцію "Use Data Bound Inems". Встановив параметри "Data Source" рівним "Other Data Source\Project Data Sources\Workers1DataSet\Workers1", а параметр "Display Member" рівний "FIO". Решта пораметри залишилися без зміни.

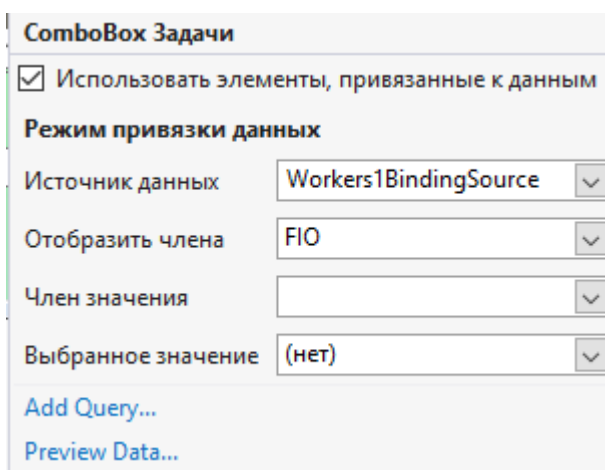


Рис.10.3- Налаштував заповнення випадаючого списку.

Зокрип вікно випадаючого списку. На понели невідомой обектів з'явитися додатковий обект зв'язку "Workers1BindingSource1", призначений для заповнення впадаючого списку. Після найстройкі всіх вищеперелічений властивостей обектів нова форма прийняла вид (ріс.10.4).

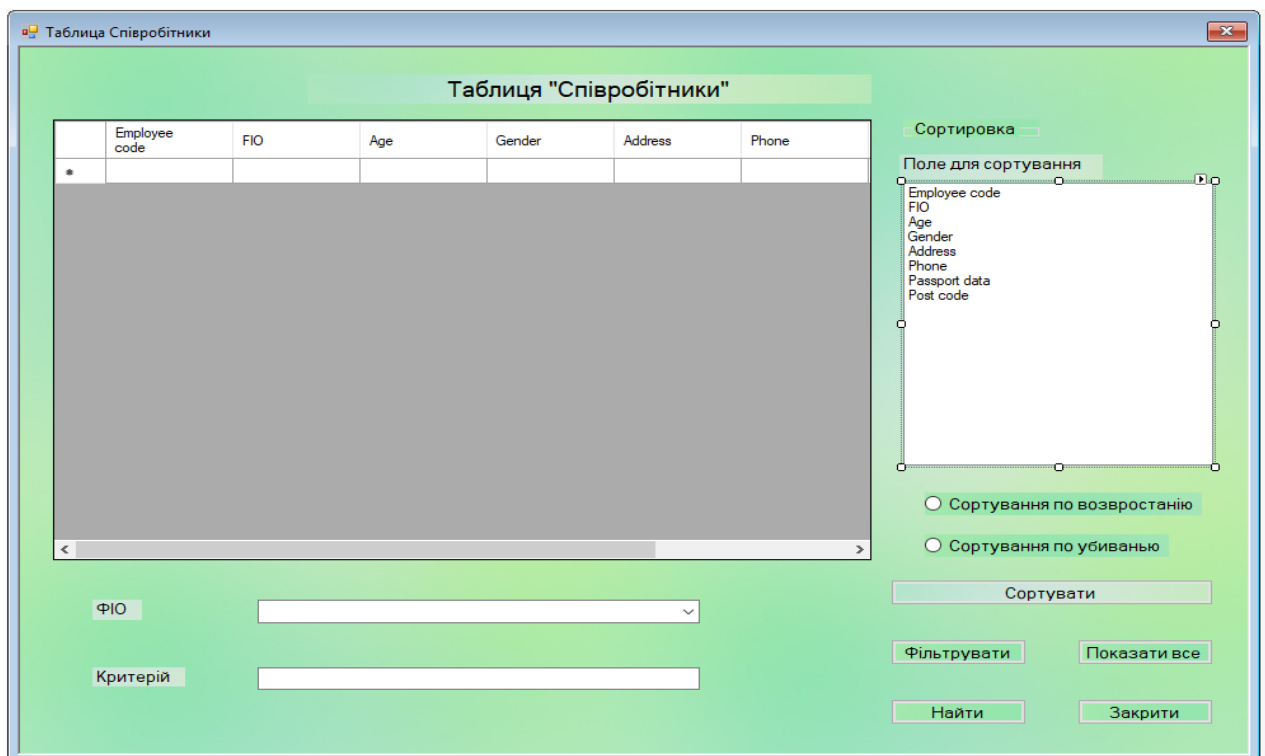


Рис.10.4-створення таблиці 'Співробітники'

На цьому ми закінчуємо настройку властивостей об'єктів і переходимо до написання коду обробників подій об'єктів.

Роботу з кодом почнемо з написання коду для розблокування кнопки "Сортувати", при виборі пункту списку (ListBox1). Для створення процедури події двічі клацніть ЛКМ по списку. З'явиться процедура обробки події, що відбувається ходить при виборі пункту списку (ListBox1.SelectedIndexChanged). У процедурі наберіть команду розблокування кнопки "Сортувати" (Button1): Button1.Enabled = True;

```
Private Sub ListBox1_SelectedIndexChanged(sender As Object, e As EventArgs) Handles ListBox1.SelectedIndexChanged
    Button1.Enabled = True
End Sub
```

Тепер перейдемо до створення коду сортування нашої таблицю в залежності від обраного поля і порядку сортування при натисканні кнопки "Сортувати". Двічі щокнуул ЛФМ по кнопці "Сортувати". З'явиться процедура "Button1 Click", здійснення при натисканні ЛКМ по кнопці. У процедурі наберіть код:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
```

```

Dim col As System.Windows.Forms.DataGridColumn
Select Case ListBox1.SelectedIndex
    Case 0
        col = DataGridViewTextBoxColumn2
    Case 1
        col = DataGridViewTextBoxColumn3
    Case 2
        col = DataGridViewTextBoxColumn4
    Case 3
        col = DataGridViewTextBoxColumn5
    Case 4
        col = DataGridViewTextBoxColumn6
    Case 5
        col = DataGridViewTextBoxColumn7
End Select
If RadioButton1.Checked Then
    ContributionsDataGridView.Sort(col,
System.ComponentModel.ListSortDirection.Ascending)
Else
    ContributionsDataGridView.Sort(col,
System.ComponentModel.ListSortDirection.Descending)
End If
End Sub

```

Для активації кнопки 'Фільтрувати' введіть наступний код:

```

Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
    ContributorsBindingSource.Filter = "[Name of the deposit]=" & ComboBox1.Text
    & ""

End Sub

```

Для активації кнопки 'Показати все' введіть наступний код:

```

Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    ContributorsBindingSource.Filter = ""

End Sub

```

Зауважимо, що якщо привласнити властивості "Filter" значення порожнього рядка (""), то його дія буде скасовано.

Далі розглянемо реалізацію пошуку інформації в таблиці. Двічі клацніть! по кнопці "Знайти". У що з'явилася процедурі обробки натиснення кнопки "Button4Click" наберіть наступний код:

```
Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    For i = 0 To ContributionsDataGridView.ColumnCount - 1
        For j = 0 To ContributionsDataGridView.RowCount - 1
            ContributionsDataGridView.Item(i, j).Style.BackColor = Color.White
            ContributionsDataGridView.Item(i, j).Style.ForeColor = Color.Black
        Next j
    Next i
    For i = 0 To ContributionsDataGridView.ColumnCount - 1
        For j = 0 To ContributionsDataGridView.RowCount - 1
            If InStr(ContributionsDataGridView.Item(i, j).Value, TextBox1.Text) Then
                ContributionsDataGridView.Item(i, j).Style.BackColor = Color.Aqua
                ContributionsDataGridView.Item(i, j).Style.ForeColor = Color.Blue
            End If
        Next j
    Next i
End Sub
```

Активація кнопки 'Закрити':

```
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    Me.Close()
End Sub
```

Створимо кнопку, яка буде зв'язувати табличну форму з стрічковою. Для цього на формі з таблицею 'Вклади' створіть кнопку та введіть код:

```
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    Form12.Show()
End Sub
```

Form6

Вклади

Deposit code:

Name of the deposit:

Minimum deposit term:

Minimum deposit amount:

Currency code:

Interest rate:

Additional conditions:

Добавить

Сохранить

Удалить

Таблица

⏮ ⏪ ⏩ ⏭

Рис.10.5-створення кнопки таблиця.

Перевірте роботу табличної форми та аналогічно зробіть з іншими таблицями.

11. Звіти

Почнемо створення стрічкового звіту, яке відображає таблицю ‘Працівники’. Створимо пустий звіт. Для цього відкрийте ‘Проект’ на панелі об’єктів-‘Добавить новый элемент’- ‘Отчёты’-‘Отчёт’-‘Добавить’.

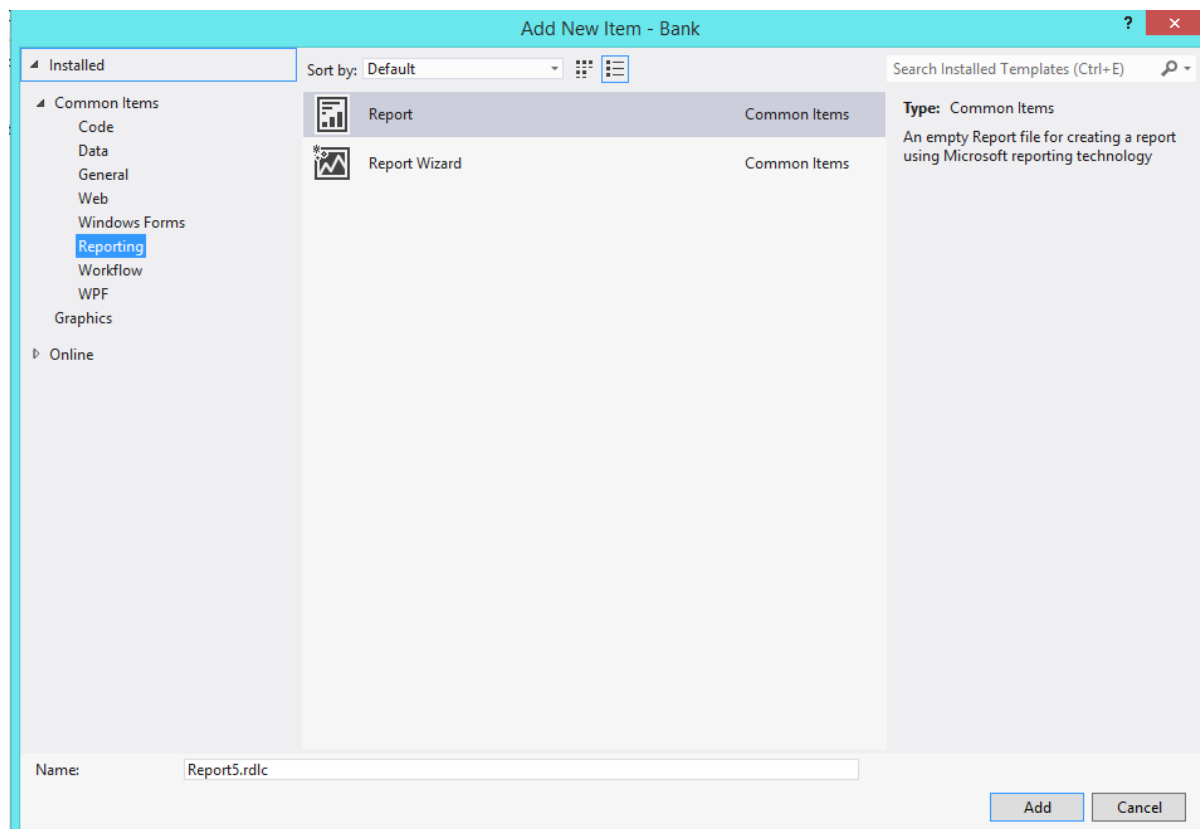


Рис.11.1- Додання форми «Report».

Також зміниться ‘Оглядач вирішень’.

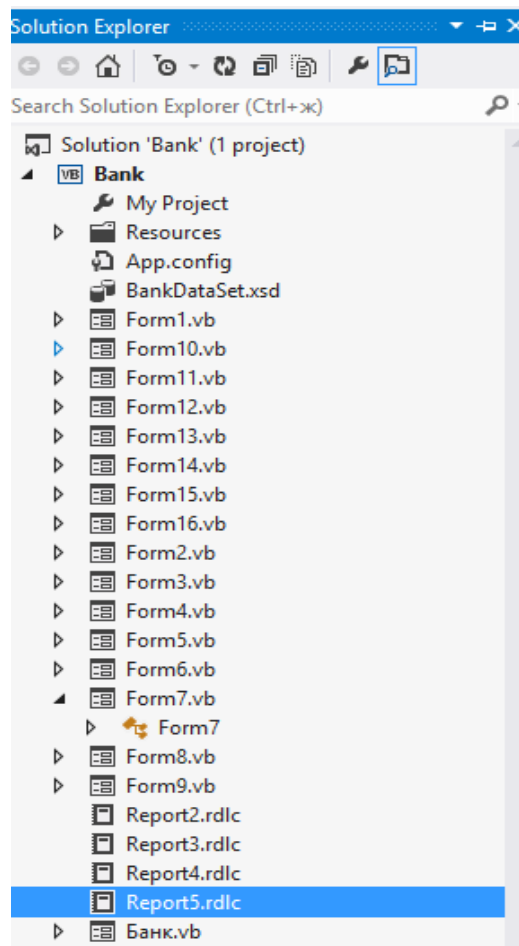


Рис.11.3-Додання звіту до списку.

Додаємо в звіт елемент ‘Таблиця’ з панелі елементів.

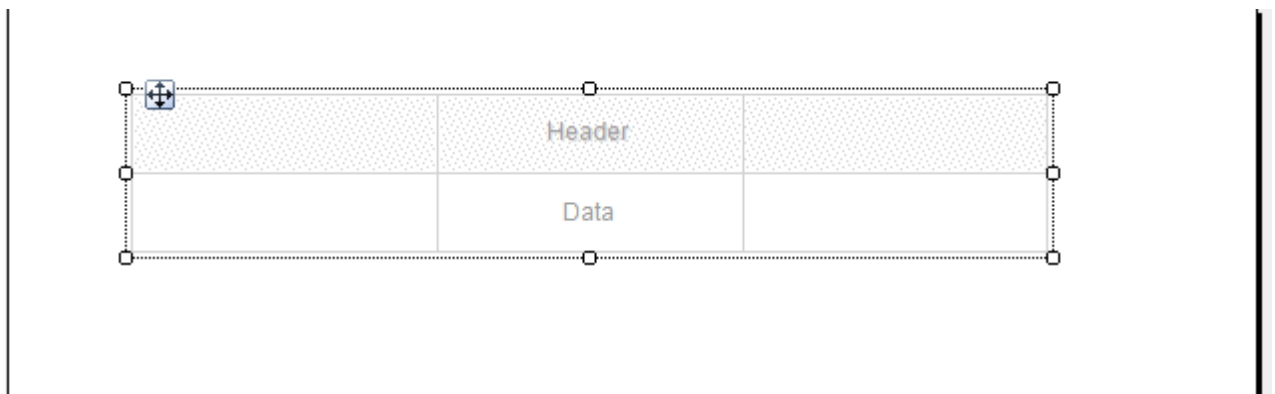


Рис.11.4-Таблиця.

Видалимо непотрібний перші стовпець, тому що для відображення даних нам потрібно лише 2 стовпця – перший для відображення полів ‘Працівники’, а другий-підписів до них. Клацніть ПКМ по заголовку

першого стовбця и виберіть ‘Удалить строки’. Також об’єднаємо ячейки. Тепер помістимо в таблицю поля таблиці ‘Вакансии’. Для цього перетягніть поля таблиці с панелі ‘Источники данных’ в ячейки праворуч. В лівому стовбці задайте імена полів. Задайте фоновий колір стовбцям(на панелі властивостей в властивості ‘BackColor’ оберіть сірий колір) і ввімкніть відображення границь ячеек (виділивши всі ячейки с полями, натисніть кнопку, яка з’явиться вище таблиці).

	Отчёт тоблицы "Вакансии"	
≡	Name of the post	«Expr»
≡	Salary	[Salary]
≡	Duties	[Duties]
≡	Requiremqnts	[Requirements]

Рис.11.5-Вигляд заповненого звіту «Вакансии».

Тепер створил форму, яка буде відображувати наш звіт. Також помістимо Label, де Text= ‘Працівники’.

Далі помістіть на форму спеціальний об’єкт звіту ‘MicrosoftReportViewer’.

Під’єднайте створений звіт(в випадаючому списку оберіть звіт ‘Report1.rdlc’.

Після всіх дій, форма буде має вигляд:

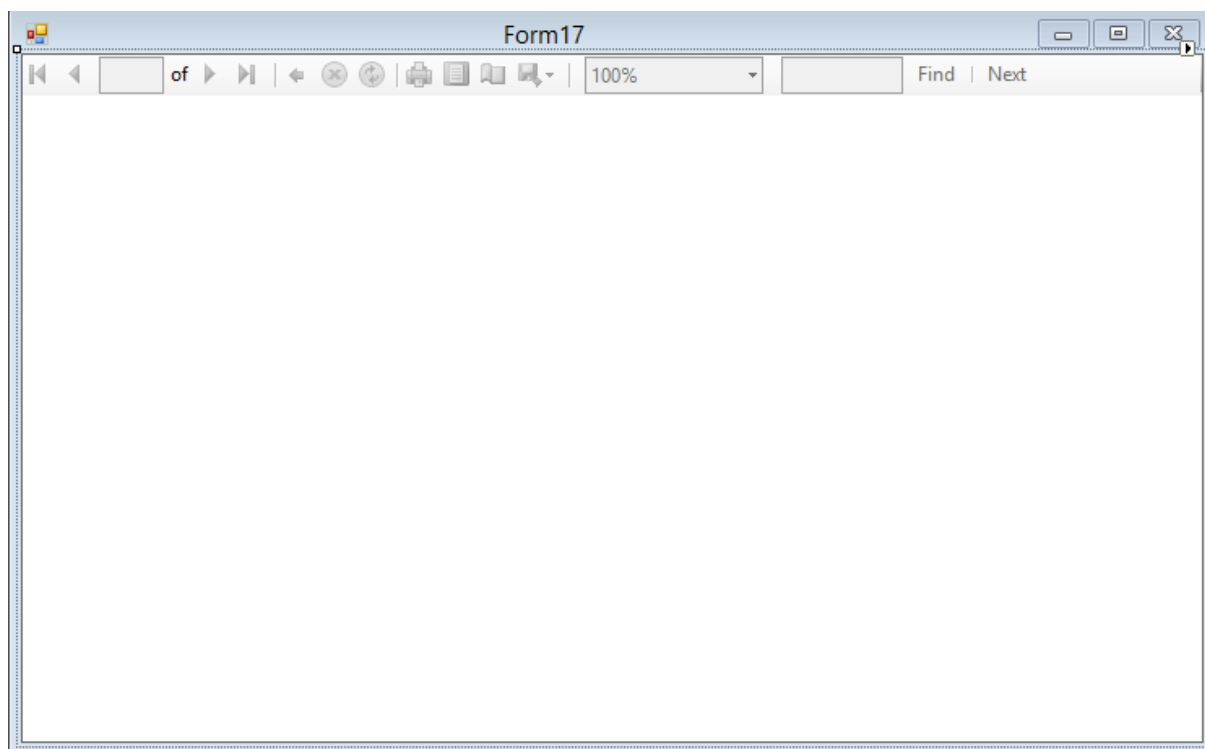


Рис.11.6-Форма для таблиці.

Під'єднайте звіт таблиці 'Вакансии' до самої таблиці.

Додайте кнопку на головній формі таблиці 'Вакансии' та ввів код переходу на звіт:

```
Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles Button2.Click
    Form14.Show()
End Sub
End Class
```

Рис.11.7-Код для переходу на звіт «Вкладники».

За аналогією створив необхідні форми та звіти.

Отчёт таблицы "Вкладники"	
FIO the depositor	«Expr»
Address	[Address]
Phone	[Phone]
Pasport data	«Expr»
Date of deposit	«Expr»
Date of return	«Expr»
Deposit code	«Expr»
The amount of the deposit	«Expr»
Amount of refund	«Expr»
Mark of return of deposit	«Expr»
Employee code	«Expr»

Рис.11.8- Вигляд заповненого звіту «Вкладники».

Отчёт таблицы "Роботники"	
FIO	[FIO]
Age	[Age]
Gender	[Gender]
Address	[Address]
Phone	[Phone]
Pasport date	«Expr»
Post code	«Expr»

Рис.11.9-
Вигляд
заповненого
звіту
«Роботники».

Отчёт таблицы "Вклады"	
Name of the deposit	«Expr»
Minimum deposit term	«Expr»
Minimum deposit amou	«Expr»
Currency code	«Expr»
Interest rate	«Expr»
Additional contrions	«Expr»

Змн.	Арк.	№ докул

	Арк.
	78

Рис.11.10- Вигляд заповненого звіту «Вклади».

Отчёт таблицы "Роботники"

Звіт "Workers"

Employee code	1
FIO	Броварчук Людмила Євгенович
Age	42
Gender	Жін
Address	40388, Івано-Франківська область, місто Івано-Франківськ, пл. Леніна, 13
Phone	+380994986711
Pasport data	ЄД-0001021
Post code	1

Отчёт таблицы "Вакансии"

Звіт "Positions"

Post code	1
Name of the post	8500 грн.
Salary	Касир-операціоніст
Duties	Касир-операціоніст - обличчя банку. Він першим зустрічає відвідувачів. До його обов'язків входить спілкування з клієнтами, контроль за станом рахунків, внесення змін в базу даних при надходженні і знятті грошей.
Requirements	великий тягар відповідальності; чималі психологічні навантаження, які виникають з постійної роботи з людьми;

Рис.11.11-пработа звіту “Workers”

Рис.11.12-пработа звіту “Positions”

Отчёт таблицы "Валюта"

Звіт "Currency"

Currency code	1
Name	UAH - гривня
Fussy course	26.90 UAH = 1 USD
Currency code 2	
Name	USD -долар
Fussy course	1 USD = 26.90 UAH

Отчёт таблицы "Владники"

Звіти "Contributors"

FIO the depositor	Анна Олексійович Кравчук
Address	72348, Донецька область, місто Донецьк, пров. Тараса Шевченка, 22
Phone	0636609769
Pasport data	MH178647
Data of deposit	25.02.2019 0:00:00
Dste of return	25.05.2019 0:00:00
Deposit code	121
The amount of the deposit	5 000 UAH
Amount of refund	5 182,5 UAH
Mark of return of deposit	Повернено
Employee code	1

Змн. | Арк. | № докум. | Підпис | Дата

жк.

9

Рис.11.13-пработа звіту “Currency”

Рис.11.14-пработа звіту “Contributors”

Висновок

В ході роботи була розроблена база даних для предметної області «Банк».

База даних була виконана в системі управління базами даних MySQL за допомогою SQL-запитів. В ході створення структура бази даних була перевірена за допомогою запитів (запит на вибірку, запит на об'єднання декількох таблиць, запит з підзапитом).

База даних готова для використання, її можна буде використовувати для розробки програмного продукту.

					5.151.1.42.08-ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

ДЖЕРЕЛА:

1. Дейт К. Дж. Введение в системы баз данных. – М.: Вильямс, 2008.
2. Уолтер Р.Э., Коулс М., Рей Р., Феррачати Ф., Фармер Д. SQL Server 2008: ускоренный курс для профессионалов. – М.: Издательский дом «Вильямс», 2008.
3. Жилинский А. Самоучитель Microsoft SQL Server 2008. – СПб.: БХВ-Петербург, 2009.
4. Lobel L., Brust A. J., Forte S. Programming Microsoft SQL Server 2008. – Microsoft Press, 2008.
5. Вьейра Р. SQL Server 2000 Программирование. Ч. 1. – М.: Изд-во

					5.151.1.42.08-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		81

«БІНОМ. Лабораторія знань», 2004.

6.[https://ru.wikipedia.org/wiki/Microsoft_SQL_Server#SQL_Server_2008_\(2005%E2%80%942008\)](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server#SQL_Server_2008_(2005%E2%80%942008))

					5.151.1.42.08-ПЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		